Iterative learning control methods for nonlinear systems

DIPLOMA THESIS

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Univ.-Prof. Dipl.-Ing. Dr. techn. Andreas Kugi
Dipl.-Ing. Dr. techn. Andreas Deutschmann-Olek BSc

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
Raphael Buchinger
Matriculation number e1630035

Vienna, December 2024

Complex Dynamical Systems Group
A-1040 Wien, Gußhausstr. 27–29, Internet: https://www.acin.tuwien.ac.at

Preamble

First and foremost, I express my deepest gratitude to my academic supervisors, Prof. Dr. Andreas Kugi and Dr. Andreas Deutschmann-Olek, for their valuable advice and insightful feedback during the writing of this thesis. Their support was crucial in shaping this thesis, particularly their excellent teaching that deepened my understanding of the topic and inspired me to delve deeper. It was a privilege to learn from them.

A heartfelt thank you to my friends and colleagues who shared this journey with me and helped me to stay balanced and focused with encouraging words, stimulating discussions and moments of laughing together. Working with you has made this experience truly memorable.

I am also very grateful to all my fellow students who have become excellent friends. Thank you for being who you are and for the times of learning and celebrating together. I will never forget the memories of our time together.

Last but certainly not least, I want to express my deepest gratitude to my parents and family. Your unwavering love, encouragement, patience, and the gift of a carefree childhood have been the foundation for this journey. Thank you for always believing in me and supporting me tirelessly. For these reasons, I dedicate this work to you.

Vienna, December 2024

Abstract

This thesis deals with the extension of iterative learning control strategies to nonlinear system classes, especially to the class of systems with affine input. It is well known from the literature that iterative learning control strategies improve the performance of repetitive processes by learning from previous trials. Therefore, this control strategy is particularly interesting when dealing with inevitable model-plant mismatches or repetitive disturbances. The extension of these strategies to nonlinear systems poses significant challenges due to their complexity, which are addressed in this thesis.

In particular, it is investigated how differential flatness, a special property of certain nonlinear systems, can be used for this purpose. This approach enables the transformation of system dynamics into an alternative flat representation, which facilitates the design of ILC laws by providing a direct relationship between output and input spaces. Furthermore, observer-based techniques are integrated into the control strategy to overcome the challenge of unmeasured state variables. These observers estimate the required system states in real-time and ensure accurate updates of the control input during the learning process.

The ILC approach proposed in this thesis, which combines differential flatness with learning concepts, provides a systematic method to improve trajectory following control for nonlinear systems. The effectiveness of these methods is demonstrated through numerical simulations and experimental validation, showing their ability to achieve high precision and convergence where traditional control approaches reach their limits.

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der Erweiterung von iterativ lernenden Regelungsstrategien auf nichtlineare Systemklassen, speziell auf die Klasse der Systeme mit affinem Eingang. Aus der Literatur ist bekannt, dass iterativ lernende Regelungsstrategien die Performance von sich wiederholenden Prozessen durch Lernen von vorherigen Versuchen verbessern. Daher ist diese Regelungsstrategie ganz besonders interessant, wenn man unvermeidbare Modellfehler oder sich wiederholende Störungen handhaben will. Die Ausweitung dieser Strategien auf nichtlineare Systeme stellt aufgrund der Komplexität dieser Systeme erhebliche Herausforderungen dar, mit welchen sich diese Arbeit beschäftigt.

Ganz speziell wird untersucht, wie differenzielle Flachheit, eine besondere Eigenschaft bestimmter nichtlinearer Systeme, für diesen Zweck genutzt werden kann. Diese Betrachtungsweise ermöglicht die Umwandlung der Systemdynamik in eine alternative flache Darstellung, welche die Gestaltung von ILC-Gesetzen erleichtert, indem sie eine direkte Beziehung zwischen Ausgangsgrößen und Eingangsgrößen darstellt. Um weiters die Herausforderung nicht gemessener Zustandsgrößen zu bewältigen, werden beobachterbasierte Techniken in die Regelungsstrategie integriert. Diese Beobachter schätzen die erforderlichen Systemzustände in Echtzeit und gewährleisten während des Lernprozesses genaue Aktualisierungen der Eingangsgrößen.

Der in dieser Arbeit vorgeschlagene ILC-Ansatz, der Differentielle-Flachheit mit lernenden Konzepten kombiniert, bietet eine systematische Methode zur Verbesserung der Trajektorienfolgeregelung für nichtlinearen Systemen. Die Wirksamkeit der Methode wird durch numerische Simulationen und experimentelle Validierung nachgewiesen und zeigt ihre Fähigkeit, hohe Präzision und Konvergenz zu erreichen, wo traditionelle Regelungsansätze an ihre Grenzen stoßen.

Contents

1 Introduction

Iterative learning mirrors how humans improve skills through repetition and adaptation but applied to machines. In human learning, skills are improved by repeating tasks, identifying recurring mistakes, and making adjustments. Similarly, in iterative learning control (ILC), the aim is to achieve specific goals in repetitive systems, such as improving tracking accuracy, by learning from the errors made in each trial. Concerning [1–3] ILC has been a topic of interest in control theory literature for over 40 years, where the academic origin of the basic concept of ILC, as formulated by [4], is to obtain "an iterative betterment process for the dynamics of robots so that the trajectory of their motion approaches asymptotically a given desired trajectory as the number of operation trials increases". Initially driven by the robotics community, ILC research has expanded to encompass various domains. While robotics applications remain a significant focus, ILC has found utility in fields such as chemical batch processing [5, 6], power electronics [7, 8], and bioengineering [9, 10] and many more which can be found in [3, 11]. However since many industrial and physical processes operate under repetitive conditions, performing the same tasks over and over again. This repetition allows the use of knowledge from previous iterations to improve the performance. In order to formulate this idea in a mathematical sense, certain requirements must be placed on the system and the repetitive process. Therefore, the following requirements are typically made [1, 2]

Assumption 1 (Applicability conditions)**.**

(i) Each trail has a fixed interval $t \in [0, T]$.

(ii) The desired trajectory $y_d(t)$ is given a priori over the fixed interval $t \in [0, T]$.

(iii) The initial conditions of the system are identical at each iteration $\mathbf{x}_j(0) = \mathbf{x}_0 \in \mathbb{R}^n, \forall j \in [0, N]$.

(iv) The dynamic of the system is invariant over all iterations.

However, as described in [2], the identical initial condition assumption (iii) is practically difficult to achieve and therefore often relaxed in the sense of

Assumption 2 (Relaxed initial condition)**.**

(i) The initial conditions at each iteration are nearly identical such that $\|\mathbf{x}_j(0) - \mathbf{x}_0\| < \varepsilon \in \mathbb{R}, \forall j \in [0, N]$ and a small ε. .

The aim of Chapter 2 is to introduce the main ideas of iterative learning, which are well known from the literature, for linear discrete-time systems. Here, the main concepts are taken from [1, 2, 12]. In Chapter 3, the first part is to extend the ideas from Chapter 2 to linear continuous-time systems. Here, the main idea is taken from [12] and has been treated with the methods of [13, 14]. Afterwards, a new type of ILC is introduced, using differential geometric and optimal control methods. In Chapter 4, the methods presented in Chapter 3 are extended to a special class of nonlinear systems called affine input systems. Finally, the method developed in Chapter 4 is applied to a three degrees-of-freedom (DOF) laboratory helicopter from the company Quanser. This system is particularly challenging from an ILC perspective due to its strong nonlinearity, underactuated nature and the significant model-plant mismatch introduced due to model reduction methods required to construct a differentially flat output.

2 ILC for discrete-time linear systems

This chapter briefly summarizes the state of the art for a single input, single output (SISO) discrete-time linear time-invariant (LTI) system in the state space representation

$$\mathbf{x}_j[k+1] = \boldsymbol{\Phi}\mathbf{x}_j[k] + \boldsymbol{\gamma}u_j[k] \ , \quad \mathbf{x}_j[0] = \mathbf{x}_0 \tag{2.1a}$$

$$y_j[k] = \mathbf{c}^{\mathrm{T}}\mathbf{x}_j[k] \ , \tag{2.1b}$$

where $\mathbf{x}_j[k+1]$ denotes the state for the j-th iteration with the initial condition $\mathbf{x}_j[0] = \mathbf{x}_0$, $u_j[k]$ the control input and $y_j[k]$ the output for all $j \in \mathbb{N}_0$. Here $\boldsymbol{\Phi} \in \mathbb{R}^{n\times n}$ stands for the dynamic (or system) matrix, $\boldsymbol{\gamma} \in \mathbb{R}^n$ for the input vector and $\mathbf{c} \in \mathbb{R}^n$ for the output vector. The main parts of this chapter follow the presentation in [1, 2, 12] with the aim to emphasize the basic idea of existing ILC strategies.

Remark 1. *This chapter only covers the SISO case for the sake of notational simplicity. Still, the ILC strategies presented in this chapter are not restricted to the SISO case and can be extended to the MIMO case.*

2.1 Lifted system representation

Based on (2.1), the lifted system representation can be introduced, which is, in principle, a mathematical technique that transforms the traditional time-domain view of a dynamic system into a higher-dimensional space [2]. Through this transformation, the entire iteration of a repetitive process can be treated as an input-output relationship. Starting from the initial state $\mathbf{x}_0$, the input-output representation of (2.1) follows as

$$y_j[k] = \mathbf{c}^{\mathrm{T}}\boldsymbol{\Phi}^k\mathbf{x}_j[0] + \mathbf{c}^{\mathrm{T}}\sum_{m=0}^{k-1}\boldsymbol{\Phi}^{k-m-1}\boldsymbol{\gamma}u_j[m] \ , \quad k = 0, 1, \dots \ . \tag{2.2}$$

The second part of (2.2) describes the convolution of the input with the convolution kernel

$$g[k] = \mathbf{c}^{\mathrm{T}}\boldsymbol{\Phi}^{k-1}\boldsymbol{\gamma} \ , \quad k = 1, 2, \dots \ , \tag{2.3}$$

which is also known as the Markov parameters of the system (2.1) [12]. Therefore, (2.2) can be rewritten as

$$y_j[k] = \mathbf{c}^{\mathrm{T}}\boldsymbol{\Phi}^k\mathbf{x}_j[0] + \sum_{m=0}^{k} g[m]u_j[k-m] \ , \quad k = 0, 1, \dots \ . \tag{2.4}$$

By considering a finite time horizon $t = [0, (N-1)T_s]$ and introducing the vector notion

$$\mathbf{u}_j^{\mathrm{T}} = [u_j[0] \quad u_j[1] \quad \dots \quad u_j[N-1]] \in \mathbb{R}^N \tag{2.5a}$$
$$\mathbf{y}_j^{\mathrm{T}} = [y_j[m] \quad y_j[m+1] \quad \dots \quad y_j[m+N-1]] \in \mathbb{R}^N \tag{2.5b}$$
$$\mathbf{y}_0^{\mathrm{T}} = [y_0[m] \quad y_0[m+1] \quad \dots \quad y_0[m+N-1]] \in \mathbb{R}^N \ , \tag{2.5c}$$

the input-output representation (2.2) can be written in the lifted system representation

$$\mathbf{y}_j = \mathbf{y}_0 + \mathbf{G}\mathbf{u}_j \ , \tag{2.6}$$

with

$$\mathbf{G} = \begin{bmatrix} g[m] & 0 & \dots & 0 \\ g[m+1] & g[m] & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ g[m+N-1] & g[m+N-2] & \dots & g[m] \end{bmatrix} . \tag{2.7}$$

The lower triangular structure of the Toeplitz matrix **G** stems from the causal nature of the dynamic system (2.1) and (2.2).

2.2 Proportional-derivative type ILC

One of the simplest ILC strategies is the PD-type, which can be written as

$$u_{j+1}[k] = u_j[k] + k_p e_j[k] + k_d(e_j[k+1] - e_j[k]) \ , \tag{2.8}$$

with $e_j[k] = y_d[k] - y_j[k]$ denoting the output error, for the j-th iteration [15]. Since this law only introduces two tuning parameters, k_p and k_d, it is considered a model-free ILC variant [1].

2.3 Filter-based ILC

Using the lifted system representation and introducing the vector form of the error

$$\mathbf{e}_j = \mathbf{y}_d - \mathbf{y}_j \tag{2.9a}$$
$$\mathbf{y}_d^{\mathrm{T}} = [y_d[0] \quad y_d[1] \quad \dots \quad y_d[N-1]] \in \mathbb{R}^N \ , \tag{2.9b}$$

the PD-type update law (2.8) can be extended to

$$\mathbf{u}_{j+1} = \mathbf{Q}(\mathbf{u}_j + \mathbf{L}\mathbf{e}_j) \ , \tag{2.10}$$

with a general **Q**-filtering matrix

$$\mathbf{Q} = \begin{bmatrix} q[0] & q[-1] & \dots & q[-(N-1)] \\ q[1] & q[0] & \dots & q[-(N-2)] \\ \vdots & \dots & \ddots & \vdots \\ q[N-1] & q[N-2] & \dots & q[0] \end{bmatrix} \tag{2.11}$$

and the learning matrix

$$\mathbf{L} = \begin{bmatrix} l[0] & l[-1] & \dots & l[-(N-1)] \\ l[1] & l[0] & \dots & l[-(N-2)] \\ \vdots & \dots & \ddots & \vdots \\ l[N-1] & l[N-2] & \dots & l[0] \end{bmatrix} . \tag{2.12}$$

Based on (2.8), the learning matrix $\mathbf{L}$ can be related with k_p and k_d as

$$\mathbf{L} = \begin{bmatrix} k_p - k_d & k_d & 0 & \dots & 0 \\ 0 & k_p - k_d & k_d & \dots & 0 \\ \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & k_p - k_d \end{bmatrix} . \tag{2.13}$$

Hence, $\mathbf{Q}$ is typically selected as a zero-phase low-pass filter to avoid strong steady-state errors due to additive phase shifts [2]. Interestingly, neither $\mathbf{Q}$ nor $\mathbf{L}$ require a lower triangular structure. This flexibility allows ILC to demonstrate a unique capability to deal with repetitive disturbances.

2.4 Inversion-based ILC

Intuitively, it is not clear *a priori* how to choose the learning matrix $\mathbf{L}$. A commonly used choice leads to the so-called inversion-based ILC [2]. Here, an inverse mapping of (2.6)

$$\mathcal{G}^{-1} : \mathcal{Y} \to \mathcal{U} \tag{2.14}$$

needs to be found. This inverse mapping allows a more direct and intuitive understanding of how input adjustments relate to output errors. A simple choice of the learning matrix $\mathbf{L}$ could be $\mathbf{G}^{-1}$ by the use of the lifted system representation of the system (2.6), which leads to the update law

$$\mathbf{u}_{j+1} = \mathbf{Q}(\mathbf{u}_j + \mathbf{G}^{-1}\mathbf{e}_j) . \tag{2.15}$$

However, this particular choice of the learning matrix has some difficulties that should not be underestimated. First, the model is never known exactly, which can lead to large errors of the inverse and unstable behaviour of the ILC. Second, $\mathbf{G}^{-1}$ is problematic for non-minimum phase systems, and third, $\mathbf{G}$ could be ill-conditioned and therefore lead to numerical difficulties. Hence, a popular choice is to regularize the system inversion by assigning

$$\mathbf{L} := (\alpha\mathbf{E} + \mathbf{G})^{-1} , \tag{2.16}$$

where $\alpha > 0$ is a small regularization parameter [2].

2.5 Norm-optimal ILC

Norm-optimal ILC eliminates the need to explicitly design a $\mathbf{Q}$-filter and a learning filter $\mathbf{L}$ that are typically required in traditional ILC approaches [12]. Here, the objective function is formulated as

$$\min_{u_{j+1}} J(u_{j+1}) = \min_{u_{j+1}} \frac{1}{2}\mathbf{e}_{j+1}^{\mathrm{T}}\mathbf{V}\mathbf{e}_{j+1} + \frac{1}{2}\mathbf{u}_{j+1}^{\mathrm{T}}\mathbf{S}\mathbf{u}_{j+1} + \frac{1}{2}(\mathbf{u}_{j+1} - \mathbf{u}_j^{\mathrm{T}})\mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j) \tag{2.17a}$$

$$\text{s.t.} \quad \mathbf{e}_{j+1} = \mathbf{y}_d - \mathbf{y}_{j+1} = \mathbf{e}_j + \mathbf{G}\mathbf{u}_j - \mathbf{G}\mathbf{u}_{j+1} \ , \tag{2.17b}$$

with the positive definite matrices **V**, **S** and **R**. By using the first-order optimality condition

$$\left(\frac{\partial J}{\partial \mathbf{u}_{j+1}}(\mathbf{u}_{j+1})\right)^{\mathrm{T}} = (\mathbf{G}^{\mathrm{T}}\mathbf{V}\mathbf{G} + \mathbf{S} + \mathbf{R})\mathbf{u}_{j+1} - (\mathbf{G}^{\mathrm{T}}\mathbf{V}\mathbf{G} + \mathbf{R})\mathbf{u}_j - \mathbf{G}^{\mathrm{T}}\mathbf{V}\mathbf{e}_j \overset{!}{=} \mathbf{0} \ , \tag{2.18}$$

the update law is written equivalently to the linear ILC (2.10) with

$$\mathbf{Q} = (\mathbf{G}^{\mathrm{T}}\mathbf{V}\mathbf{G} + \mathbf{S} + \mathbf{R})^{-1}(\mathbf{G}^{\mathrm{T}}\mathbf{V}\mathbf{G} + \mathbf{R}) \tag{2.19a}$$

$$\mathbf{L} = (\mathbf{G}^{\mathrm{T}}\mathbf{V}\mathbf{G} + \mathbf{R})^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{V} \ . \tag{2.19b}$$

3 ILC for continuous-time linear systems

In comparison to Chapter 2, this chapter considers a single input, single output (SISO) continuous-time LTI system of the form

$$\dot{\mathbf{x}}_j(t) = \mathbf{A}\mathbf{x}_j(t) + \mathbf{b}u_j(t) \ , \quad \mathbf{x}_j(0) = \mathbf{x}_0 \tag{3.1a}$$

$$y_j(t) = \mathbf{c}^{\mathrm{T}}\mathbf{x}_j(t) \ , \tag{3.1b}$$

where $\mathbf{x}_j(t)$ denotes the state for the j-th iteration with the initial condition $\mathbf{x}_j(0) = \mathbf{x}_0$, $u_j(t)$ the control input and $y_j(t)$ the output for all $j \in \mathbb{N}_0$. Here $\mathbf{A} \in \mathbb{R}^{n\times n}$ stands for the dynamic (or system) matrix, $\mathbf{b} \in \mathbb{R}^n$ for the input vector and $\mathbf{c} \in \mathbb{R}^n$ for the output vector. This introductory chapter lays the foundations for the methods developed in a more general framework in the following chapter.

Remark 2. *This chapter focuses exclusively on the SISO (single input, single output) case to maintain notational simplicity. This choice enhances clarity, facilitates a solid foundation for understanding core concepts, reduces mathematical complexity, and allows easier visualization. Still, the ILC strategies presented in this chapter are not restricted to the SISO case and can be extended to the MIMO case. For the same reason, the time argument t is omitted when the context implies its presence.*

3.1 LQT-based ILC

One main disadvantage of the norm-optimal ILC is that it uses the lifted system representation and, therefore, it is computationally expensive for increasing trail lengths T or decreasing sampling times T_s [12]. Furthermore, unstable plants must be stabilized before applying ILC methods of the form (2.10). These disadvantages can be bypassed by deriving the ILC update law from a linear quadratic tracking (LQT) problem [12]. Therefore, the optimization problem, with the positive weights F, Q and R, can be written as

$$\min_{v_{j+1}(\cdot)} J(v_{j+1}) = \frac{1}{2}(e_{j+1}Fe_{j+1})\big|_{t=T} + \frac{1}{2}\int_0^T e_{j+1}Qe_{j+1} + v_{j+1}Rv_{j+1}\,\mathrm{d}t \tag{3.2a}$$

$$\text{s.t.} \quad \dot{\mathbf{z}}_{j+1} = \mathbf{A}\mathbf{z}_{j+1} + \mathbf{b}v_{j+1} \tag{3.2b}$$

$$e_{j+1} = y_d - y_{j+1} = y_d - \mathbf{c}^{\mathrm{T}}\mathbf{x}_j + \mathbf{c}^{\mathrm{T}}\mathbf{x}_j - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{j+1} = e_j - \mathbf{c}^{\mathrm{T}}\mathbf{z}_{j+1} \tag{3.2c}$$

where $\mathbf{z}_{j+1} = \mathbf{x}_{j+1} - \mathbf{x}_j$ describes the difference over the iteration of the state and $v_{j+1} = u_{j+1} - u_j$ the difference in the control input. The Hamiltonian for the LQT-based ILC formulation is given by

$$H = \frac{1}{2}(e_jQe_j - 2\mathbf{z}_{j+1}^{\mathrm{T}}\mathbf{c}Qe_j + \mathbf{z}_{j+1}^{\mathrm{T}}\mathbf{c}Q\mathbf{c}^{\mathrm{T}}\mathbf{z}_{j+1} + v_{j+1}Rv_{j+1}) + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{A}\mathbf{z}_{j+1} + \mathbf{b}v_{j+1}) \tag{3.3}$$

Using the optimality conditions [14], it follows that

$$\frac{\partial H}{\partial v}(v_{j+1}) = R v_{j+1}^{*} + \mathbf{b}^{\mathrm{T}} \boldsymbol{\lambda}^{*} \overset{!}{=} 0 \tag{3.4a}$$

$$\left(\frac{\partial H}{\partial \mathbf{z}}(\mathbf{z}_{j+1}^{*})\right)^{\mathrm{T}} = \mathbf{c} Q \mathbf{c}^{\mathrm{T}} \mathbf{z}_{j+1}^{*} - \mathbf{c} Q e_j^{*} + \mathbf{A}^{\mathrm{T}} \boldsymbol{\lambda}^{*} \overset{!}{=} -\dot{\boldsymbol{\lambda}}^{*} \ . \tag{3.4b}$$

Due to readability, the star $(\cdot)^{*}$ for the optimal solution is dropped. A closer look at (3.4b) shows that the co-state differential equation depends on the state of the $j+1$-st iteration, which is not available for all t at the time when (3.4) has to be solved. Therefore, the Ansatz function [13] for the co-state

$$\boldsymbol{\lambda} = \mathbf{S} \mathbf{z}_{j+1} + \mathbf{g} \tag{3.5}$$

is introduced. By differentiating the Ansatz function with respect to time and setting it equal to (3.4b), it follows that

$$\dot{\boldsymbol{\lambda}} = \dot{\mathbf{S}} \mathbf{z}_{j+1} + \mathbf{S} \dot{\mathbf{z}}_{j+1} + \dot{\mathbf{g}} \overset{!}{=} -\mathbf{c} Q \mathbf{c}^{\mathrm{T}} \mathbf{z}_{j+1} + \mathbf{c} Q e_j - \mathbf{A}^{\mathrm{T}} \mathbf{S} \mathbf{z}_{j+1} - \mathbf{A}^{\mathrm{T}} \mathbf{g} \ . \tag{3.6}$$

This leads to the well-known Riccati differential equation for $\mathbf{S}$ and the feedforward differential equation for the external forcing term $\mathbf{g}$ in the form

$$\dot{\mathbf{S}} = -\mathbf{S}\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{S} + \mathbf{S}\mathbf{b}R^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{S} - \mathbf{c}Q\mathbf{c}^{\mathrm{T}} \ , \tag{3.7a}$$

$$\dot{\mathbf{g}} = (\mathbf{S}\mathbf{b}R^{-1}\mathbf{b}^{\mathrm{T}} - \mathbf{A}^{\mathrm{T}})\mathbf{g} + \mathbf{c}Q e_j \tag{3.7b}$$

with the terminal condition

$$\mathbf{S}(T) = \mathbf{c} F \mathbf{c}^{\mathrm{T}} \tag{3.8a}$$

$$\mathbf{g}(T) = -\mathbf{c} F e_j(T) \ . \tag{3.8b}$$

Inserting (3.5) into (3.4a) leads to the ILC update law

$$u_{j+1} = u_j - \underbrace{R^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{S}\mathbf{z}_{j+1}}_{\text{feedback term}} - \underbrace{R^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{g}}_{\text{feedforward term}} \ . \tag{3.9}$$

To summarize the algorithm of the LQT-based ILC, (3.7a) and (3.7b) have to be solved with the corresponding terminal conditions (3.8a) and (3.8b) in order to calculate the feedback and feedforward terms. In this formulation, the feedforward term implies the knowledge of the tracking error e_j from the previous iteration. The solution of the Riccati differential equation $\mathbf{S}$ is fixed over the iterations, and the external forcing term $\mathbf{g}$ has to be recalculated after every iteration. For the zeroth iteration, $\mathbf{x}_{-1}$,u_{-1} and y_{-1} are set to zero, and therefore, the ILC update law implies the classical LQT formulation of [13, 14].

3.2 Flatness-based ILC

As seen in the previous sections, the ideas of norm-optimal ILC in Section 2.5 naturally transfer to continuous-time systems using the LQT formulation. Since ILC strategies intrinsically try to invert the system's behaviour approximately, it seems worthwhile to investigate connections to system inversion and trajectory planning methods for nonlinear systems such as flatness-based methods [16]. In this section, these techniques are applied to linear systems and extended to nonlinear AI-systems in the following chapter. Therefore, the following definition is introduced:

Definition 3.1 (Relative degree of a linear SISO-system [16])**.** *The linear SISO-system (3.1) is said to have the relative degree r if*

(i) $\mathbf{c}^{\mathrm{T}}\mathbf{A}^{i}\mathbf{b} = 0 \;, \quad \forall i \in [0, r-2]$ *and*

(ii) $\mathbf{c}^{\mathrm{T}}\mathbf{A}^{r-1}\mathbf{b} \neq 0 \quad .$

This leads to the definition of a differentially flat system where $r = n$. What makes these systems particularly interesting is that all their system variables, including both states and input variables, can be expressed as functions of a special output variable and its time derivatives. This special output variable is called the *flat output* in this context. This requirement for the flat output $z = \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{x}$ yields the linear system of equations

$$\boldsymbol{\lambda}^{\mathrm{T}} \underbrace{\left[\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{n-2}\mathbf{b}, \mathbf{A}^{n-1}\mathbf{b}\right]}_{=\mathcal{R}(\mathbf{A},\mathbf{b})} = \underbrace{\left[0, 0, \ldots, 0, \alpha \neq 0\right]}_{=\alpha\mathbf{e}^{\mathrm{T}}} \quad . \tag{3.10}$$

If the reachability matrix $\mathcal{R}(\mathbf{A}, \mathbf{b})$ is of full rank n, it follows that

$$\boldsymbol{\lambda}^{\mathrm{T}} = \alpha\mathbf{e}^{\mathrm{T}}\mathcal{R}(\mathbf{A}, \mathbf{b})^{-1} \tag{3.11}$$

Since the definition of a flat output only depends on $\alpha \neq 0$, it follows that $z = \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{x}$ is not unique. To transform the system into *Brunovsky normal form*, one has to introduce the following linear state transformation

$$\mathbf{z} = \begin{bmatrix} z \\ \dot{z} \\ \vdots \\ z^{(n-1)} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}^{\mathrm{T}} \\ \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A} \\ \vdots \\ \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}^{n-1} \end{bmatrix} \mathbf{x} = \mathbf{T}\mathbf{x} \; . \tag{3.12}$$

Therefore, the linear system from (3.1) transforms to

$$\begin{aligned} z^{(n)} &= \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}^{n}\mathbf{T}^{-1}\mathbf{z} + \alpha u \\ \mathbf{z}(0) &= \mathbf{z}_0 = \mathbf{T}\mathbf{x}_0 \in \mathbb{R}^n \\ y &= \mathbf{c}^{\mathrm{T}}\mathbf{T}^{-1}\mathbf{z} \; . \end{aligned} \tag{3.13}$$

Hence, the states $\mathbf{x}$, the input u and an arbitrary output y can be parametrized by the flat output z and its time derivatives as

$$\begin{aligned} \mathbf{x} &= \mathbf{T}^{-1}\mathbf{z} \\ u &= \frac{1}{\alpha}\left(-\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{A}^{n}\mathbf{T}^{-1}\mathbf{z} + z^{n}\right) \\ y &= \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{T}^{-1}\mathbf{z} \; . \end{aligned} \tag{3.14}$$

This inverse system builds the base to build up a flatness-based ILC.

3.2.1 LQT for linear flat systems

First, the dynamical optimization problem

$$\begin{aligned} \min_{u(\cdot)} \quad & J(u) && (3.15a) \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u && (3.15b) \\ & y = \mathbf{c}^{\mathrm{T}}\mathbf{x} \ , && (3.15c) \end{aligned}$$

with

$$J(u) = \underbrace{\left(\frac{1}{2}\sum_{i=0}^{n-1} e^{(i)} \sum_{i=0}^{n-1} Q_{i,k} e^{(k)}\right)\Bigg|_{t=T}}_{=\varphi(\mathbf{x}(T))} + \frac{1}{2}\int_0^T \sum_{i=0}^{n-1} e^{(i)} Q_i e^{(i)} + e^{(n)} e^{(n)} \, \mathrm{d}t \tag{3.16}$$

where $Q_i > 0$ and

$$\begin{aligned} e^{(i)} &= y_d^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x} \quad , \quad \forall i \in [0, n-1] && (3.17a) \\ e^{(n)} &= y_d^{(n)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u \ , && (3.17b) \end{aligned}$$

is introduced, where the system is flat concerning the output y. The question arises if the optimization problem from (3.15) could be solved by the well-known Ansatz function [14] for the co-state

$$\boldsymbol{\lambda} = \mathbf{S}\mathbf{x} + \mathbf{g} \ . \tag{3.18}$$

By differentiating the Ansatz function (3.18) with respect to time and setting it equal to (3.39b), it follows that

$$\begin{aligned} \dot{\boldsymbol{\lambda}} &= \dot{\mathbf{S}}\mathbf{x} + \mathbf{S}\dot{\mathbf{x}} + \dot{\mathbf{g}} \\ &\overset{!}{=} \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\left(y_d^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\right) - \left(\mathbf{A}^{\mathrm{T}} - \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}}\right)(\mathbf{S}\mathbf{x} + \mathbf{g}) \ . \end{aligned} \tag{3.19}$$

This leads to the Riccati differential equation for $\mathbf{S}$ and the feedforward differential equation for the external forcing term $\mathbf{g}$, which are given in the following theorem.

Theorem 3.1 (Solution of the Riccati equation for linear fully reachable SISO systems)**.** *The solutions of the Riccati differential equation*

$$\begin{aligned} \dot{\mathbf{S}} &= -\mathbf{S}\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{S} + \left(\frac{1}{\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}}\right)^2 \mathbf{S}\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{S} \\ &\quad + \frac{1}{\mathbf{c}^{\mathrm{T}}\mathbf{A}\mathbf{b}}\left(\mathbf{S}\mathbf{b}\mathbf{c}^{\mathrm{T}}\mathbf{A}^n + (\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}}\mathbf{S}\right) - \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\mathbf{c}^{\mathrm{T}}\mathbf{A}^i \end{aligned} \tag{3.20}$$

and the feedforward differential equation

$$\begin{aligned} \dot{\mathbf{g}} &= \left(\left(\frac{1}{\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}}\right)^2 \mathbf{S}\mathbf{b}\mathbf{b}^{\mathrm{T}} + \frac{1}{\mathbf{c}^{\mathrm{T}}\mathbf{A}\mathbf{b}}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}} - \mathbf{A}^{\mathrm{T}}\right)\mathbf{g} \\ &\quad - \frac{1}{\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}}\mathbf{S}\mathbf{b}y_d^{(n)} + \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i y_d^{(i)} \ , \end{aligned} \tag{3.21}$$

are given by

$$\mathbf{S} = \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \tag{3.22a}$$

$$\mathbf{g} = \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i,k} y_d^{(i)} \tag{3.22b}$$

iff

$$Q_i + 2Q_{i,i-1} - Q_{i,n-1}Q_{n-1,i} = 0 \; , \quad \forall i \in [0, n-1] \tag{3.23a}$$

$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} = 0 \; , \quad \forall i,k \in [0, n-1] \text{ and } i \neq k \tag{3.23b}$$

$$Q_{i,k} = Q_{k,i} \; . \tag{3.23c}$$

Proof. Before beginning the proof, the following identity is introduced.

Lemma 3.2. *The following equation holds true*

$$\mathbf{b}^{\mathrm{T}} (\mathbf{A}^{n-1})^{\mathrm{T}} \mathbf{c} \equiv \mathbf{b}^{\mathrm{T}} \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \; . \tag{3.24}$$

Proof. In order to prove the identity (3.24), the first point of the relative degree Definition 3.1, i.e.,

$$\mathbf{c}^{\mathrm{T}} (\mathbf{A}^i) \mathbf{b} = 0 \, , \qquad \forall i \in [0, r-2] \; , \tag{3.25}$$

is used. Since $r = n$, it directly follows

$$\mathbf{b}^{\mathrm{T}} \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} = \sum_{i=0}^{n-1} \left(\mathbf{c}^{\mathrm{T}} \mathbf{A}^i \mathbf{b} \right)^{\mathrm{T}} = \mathbf{b}^{\mathrm{T}} (\mathbf{A}^{n-1})^{\mathrm{T}} \mathbf{c} \; . \tag{3.26}$$

□

Inserting (3.22a) into (3.20) leads to

$$
\begin{aligned}
\dot{\mathbf{S}} = &- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^{k+1} - \sum_{i=0}^{n-1} (\mathbf{A}^{i+1})^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \\
&+ \left(\frac{1}{\mathbf{c}^{\mathrm{T}} \mathbf{A}^{n-1} \mathbf{b}} \right)^2 \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \underbrace{\sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{b}}_{\overset{(3.24)}{=} Q_{i,n-1} \mathbf{c}^{\mathrm{T}} \mathbf{A}^{n-1} \mathbf{b}} \underbrace{\mathbf{b}^{\mathrm{T}} \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c}}_{\overset{(3.24)}{=} \mathbf{b}^{\mathrm{T}} (\mathbf{A}^{n-1})^{\mathrm{T}} \mathbf{c}} \sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \\
&+ \frac{1}{\mathbf{c}^{\mathrm{T}} \mathbf{A} \mathbf{b}} \left(\sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \underbrace{\sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{b}}_{\overset{(3.24)}{=} Q_{i,n-1} \mathbf{c}^{\mathrm{T}} \mathbf{A}^{n-1} \mathbf{b}} \mathbf{c}^{\mathrm{T}} \mathbf{A}^n + (\mathbf{A}^n)^{\mathrm{T}} \mathbf{c} \underbrace{\mathbf{b}^{\mathrm{T}} \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c}}_{\overset{(3.24)}{=} \mathbf{b}^{\mathrm{T}} (\mathbf{A}^{n-1})^{\mathrm{T}} \mathbf{c}} \sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \right) \\
&- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} Q_i \mathbf{c}^{\mathrm{T}} \mathbf{A}^i \\
= &- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=1}^{n} Q_{i,k-1} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k - \sum_{i=1}^{n} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i-1,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \\
&+ \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} Q_{i,n-1} \sum_{k=0}^{n-1} Q_{n-1,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \\
&+ \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} Q_{i,n-1} \mathbf{c}^{\mathrm{T}} \mathbf{A}^n + (\mathbf{A}^n)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{n-1,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \\
&- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} Q_i \mathbf{c}^{\mathrm{T}} \mathbf{A}^i \, ,
\end{aligned}
\tag{3.27}
$$

by extracting the n-th summand of k in the fourth last row, it can be seen that they are equal with the second last row, and therefore, it follows that

$$
\begin{aligned}
\dot{\mathbf{S}} = &- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=1}^{n-1} Q_{i,k-1} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k - \sum_{i=1}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i-1,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \\
&+ \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i,n-1} Q_{n-1,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k - \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} Q_i \mathbf{c}^{\mathrm{T}} \mathbf{A}^i \overset{!}{=} 0 \, .
\end{aligned}
\tag{3.28}
$$

Extending the first summation to $k = 0$ where $Q_{i,-1} = 0$ and the second one to $i = 0$ where $Q_{-1,k} = 0$ and factorizing the corresponding expressions $\mathbf{c}^{\mathrm{T}} \mathbf{A}^k$, it follows that

$$
\begin{aligned}
&- \sum_{\substack{i=0 \\ i \neq k}}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} (Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1} Q_{n-1,k}) \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \overset{!}{=} 0 \, , \\
&- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} (Q_i + Q_{i,i-1} + Q_{i-1,i} - Q_{i,n-1} Q_{n-1,i}) \mathbf{c}^{\mathrm{T}} \mathbf{A}^i \overset{!}{=} 0 \, .
\end{aligned}
\tag{3.29}
$$

For the feedforward term $\mathbf{g}$, equation (3.22b) is plugged into (3.21), which leads to

$$\begin{aligned}\dot{\mathbf{g}} = &\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,n-1}Q_{n-1,k}y_d^{(k)} - \sum_{i=1}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i-1,k}y_d^{(k)} \\ &- \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_{i,n-1}y_d^{(n)} + \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i y_d^{(i)} \overset{!}{=} \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=1}^{n} Q_{i,k-1}y_d^{(k)} \ ,\end{aligned} \tag{3.30}$$

similarly to (3.29). Moving the right-hand side of this equation to the left and applying the same extension as before, it follows that

$$\begin{aligned}&-\sum_{\substack{i=0\\ i\neq k}}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}(Q_{i,k-1}+Q_{i-1,k}-Q_{i,n-1}Q_{n-1,k})y_d^{(k)} \overset{!}{=} 0 \ , \\ &-\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}(Q_i+Q_{i,i-1}+Q_{i-1,i}-Q_{i,n-1}Q_{n-1,i})y_d^{(i)} \overset{!}{=} 0 \ .\end{aligned} \tag{3.31}$$

Therefore, from (3.29) and (3.31) it follows that (3.22a) and (3.22b) are solutions of (3.20) and (3.21) iff

$$Q_i + Q_{i,i-1} + Q_{i-1,i} - Q_{i,n-1}Q_{n-1,i} = 0 \ , \quad \forall i \in [0, n-1] \tag{3.32a}$$

$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} = 0 \ , \quad \forall i,k \in [0, n-1] \text{ and } i \neq k \tag{3.32b}$$

Since the Riccati differential equation (3.20) is symmetric, i.e., $\dot{\mathbf{S}}^{\mathrm{T}} = \dot{\mathbf{S}}$, it follows that the solution must be symmetric and therefore $Q_{i,k} = Q_{k,i}$. □

From Theorem 3.1, it could be seen that there exists a solution for the Riccati differential equation and the feedforward differential equation. Inserting these solutions in the Ansatz function (3.18) leads to

$$\begin{aligned}\boldsymbol{\lambda} &= -\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\right) \\ &= \underbrace{\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}}_{=\mathbf{S}\mathbf{x}} - \underbrace{\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}y_d^{(k)}}_{=-\mathbf{g}} \ .\end{aligned} \tag{3.33}$$

Furthermore, it could be seen that for the given optimization problem (3.15), the Riccati differential equation has to be static. This leads to the following theorem.

Theorem 3.3 (Flatness based control law for a linear system)**.** *Assume the linear SISO-system (3.1) has a relative degree $r = n$. The unique solution of the optimization problem (3.15) is given by the control law*

$$u = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x} + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\right)\right) \tag{3.34}$$

and the solution of the co-state differential equations

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\left(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{i}\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k}\mathbf{x}\right) , \tag{3.35}$$

iff

$$\begin{aligned} Q_i + 2Q_{i,i-1} - Q_{i,n-1}Q_{n-1,i} &= 0 \ , \quad \forall i \in [0, n-1] && (3.36a)\\ Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} &= 0 \ , \quad \forall i,k \in [0, n-1] \ \textit{and}\ i \neq k && (3.36b)\\ Q_{i,k} &= Q_{k,i} \ , && (3.36c)\end{aligned}$$

with $Q_i > 0$.

Proof. For the optimization problem (3.15), the Hamiltonian function follows as

$$\begin{aligned} H = \frac{1}{2}\Bigg[&\sum_{i=0}^{n-1} y_d^{(i)}Q_i y_d^{(i)} - 2\mathbf{x}^{\mathrm{T}}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i y_d^{(i)} + \mathbf{x}^{\mathrm{T}}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\\ &+ y_d^{(n)}y_d^{(n)} - 2y_d^{(n)}\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x} - 2y_d^{(n)}\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u + \mathbf{x}^{\mathrm{T}}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}\\ &+ 2\mathbf{x}^{\mathrm{T}}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u + u\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{n-1})^{\mathrm{T}}\mathbf{c}\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u\Bigg] + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{A}\mathbf{x} + \mathbf{b}u)\end{aligned} \tag{3.37}$$

For optimality [14], the following relations must hold in the interval $0 \leq t \leq T$

$$\begin{aligned} \dot{\mathbf{x}}^* &= \left(\frac{\partial H}{\partial \boldsymbol{\lambda}}(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*)\right)^{\mathrm{T}} , & \mathbf{x}^*(0) &= \mathbf{x}_0 && (3.38a)\\ \dot{\boldsymbol{\lambda}}^* &= -\left(\frac{\partial H}{\partial \mathbf{x}}(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*)\right)^{\mathrm{T}} , & \boldsymbol{\lambda}^*(T) &= \left(\frac{\partial \varphi}{\partial \mathbf{x}_1}\right)^{\mathrm{T}}(\mathbf{x}^*(T)) && (3.38b)\\ 0 &= \frac{\partial H}{\partial u}(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*) , &&&& (3.38c)\end{aligned}$$

with $\mathbf{x}_1 = \mathbf{x}(T)$. For readability, the star $(^*)$ for the optimal solution is dropped in the following. From (3.38) it follows

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u && (3.39a)\\ \dot{\boldsymbol{\lambda}} &= \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\left(y_d^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\right) - \left(\mathbf{A}^{\mathrm{T}} - \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}}\right)\boldsymbol{\lambda} && (3.39b)\\ u &= -\frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})^2}\mathbf{b}^{\mathrm{T}}\boldsymbol{\lambda} + \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x} + y_d^{(n)}\right) . && (3.39c)\end{aligned}$$

Introducing the Ansatz function for the co-state differential equation in the form

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\left(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{i}\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k}\mathbf{x}\right) , \tag{3.40}$$

has been shown to be a powerful choice since the optimization problem (3.15) can be solved analytically. To show this, this Ansatz function is inserted into (3.39b), from which it follows that

$$\begin{aligned}\dot{\boldsymbol{\lambda}} &= \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\Big(y_d^{(i)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\Big)\\ &+\Big(\mathbf{A}^{\mathrm{T}}-\frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}}\Big)\sum_{i=0}^{n-1}\Big(\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\Big)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\\ &=\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\Big(y_d^{(i)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\Big)+\sum_{i=0}^{n-1}(\mathbf{A}^{i+1})^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\\ &-\frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}}\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\ .\end{aligned}\tag{3.41}$$

By the use of Lemma 3.2, the co-state differential equation for optimality (3.41) can be further simplified to

$$\begin{aligned}\dot{\boldsymbol{\lambda}} &= \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\Big(y_d^{(i)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\Big)+\sum_{i=0}^{n-1}(\mathbf{A}^{i+1})^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\\ &-\frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\ \underbrace{\mathbf{b}^{\mathrm{T}}\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}}_{\overset{(3.24)}{=}\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{n-1})^{\mathrm{T}}\mathbf{c}}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\\ &=\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\Big(y_d^{(i)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\Big)+\sum_{i=1}^{n}(\mathbf{A}^{i})^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{i-1,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\\ &-(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{n-1,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\\ &=\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\Big(y_d^{(i)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}\Big)+\sum_{i=1}^{n-1}(\mathbf{A}^{i})^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{i-1,k}\Big(y_d^{(k)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\Big)\ .\end{aligned}\tag{3.42}$$

Differentiating (3.40) with respect to time yields

$$\dot{\boldsymbol{\lambda}} = -\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k+1)}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^{k+1}\mathbf{x}-\mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{b}u\Big)\ .\tag{3.43}$$

Inserting the control input u from (3.39c) leads to

$$\begin{aligned}
\dot{\boldsymbol{\lambda}} = &- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=1}^{n} Q_{i,k-1} \Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big) \\
&+ \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \underbrace{\sum_{k=0}^{n-1} Q_{i,k} \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{b}}_{\overset{(3.24)}{=} Q_{i,n-1} \mathbf{c}^{\mathrm{T}} \mathbf{A}^{n-1} \mathbf{b}} \Bigg(+ \frac{1}{(\mathbf{c}^{\mathrm{T}} \mathbf{A}^{n-1} \mathbf{b})^2} \underbrace{\mathbf{b}^{\mathrm{T}} \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c}}_{\overset{(3.24)}{=} \mathbf{b}^{\mathrm{T}} (\mathbf{A}^{n-1})^{\mathrm{T}} \mathbf{c}} \sum_{k=0}^{n-1} Q_{i,k} \Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big) \\
&+ \frac{1}{(\mathbf{c}^{\mathrm{T}} \mathbf{A}^{n-1} \mathbf{b})} \Big(-\mathbf{c}^{\mathrm{T}} \mathbf{A}^n \mathbf{x} + y_d^{(n)} \Big) \Bigg) \\
= &- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=1}^{n-1} Q_{i,k-1} \Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big) + \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} Q_{i,n-1} Q_{n-1,k} \Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big) \; .
\end{aligned} \tag{3.44}$$

Subtracting equation (3.42) from (3.44), the coefficients $Q_{i,k}$ required in (3.34) can be found. Therefore, the first summation of the last line in (3.44) is extended to $k = 0$ where $Q_{i,-1} = 0$ and the second summation of the last line in (3.42) to $i = 0$ where $Q_{-1,k} = 0$. By factorizing the corresponding expressions $\Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big)$, it follows that

$$- \sum_{\substack{i=0 \\ i \neq k}}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} \sum_{k=0}^{n-1} (Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1} Q_{n-1,k}) \Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big) \overset{!}{=} 0 \; , \tag{3.45a}$$

$$- \sum_{i=0}^{n-1} (\mathbf{A}^i)^{\mathrm{T}} \mathbf{c} (Q_i + Q_{i,i-1} + Q_{i-1,i} - Q_{i,n-1} Q_{n-1,i}) \Big(y_d^{(k)} - \mathbf{c}^{\mathrm{T}} \mathbf{A}^k \mathbf{x} \Big) \overset{!}{=} 0 \; , \tag{3.45b}$$

must hold and, therefore

$$Q_i + Q_{i,i-1} + Q_{i-1,i} - Q_{i,n-1} Q_{n-1,i} = 0 \; , \quad \forall i \in [0, n-1] \tag{3.46a}$$

$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1} Q_{n-1,k} = 0 \; , \quad \forall i, k \in [0, n-1] \text{ and } i \neq k \; . \tag{3.46b}$$

The next step is to check if the terminal condition is satisfied by the Ansatz function

(3.40). Therefore, the final cost term is differentiated with respect to the state, i.e.

$$\left(\frac{\partial \varphi}{\partial \mathbf{x}_1}\right)^{\mathrm{T}} = \left(\frac{\partial}{\partial \mathbf{x}_1}\left(\frac{1}{2}\sum_{i=0}^{n-1}\left(y_d^{(i)}(T) - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_1\right)\sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)}(T) - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_1\right)\right)\right)^{\mathrm{T}} \tag{3.47a}$$

$$= \frac{1}{2}\Bigg[-\sum_{i=0}^{n-1} y_d^{(i)}(T)\sum_{k=0}^{n-1} Q_{i,k}\left(\mathbf{A}^k\right)^{\mathrm{T}}\mathbf{c} \tag{3.47b}$$

$$-\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)}(T) - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_1\right) \tag{3.47c}$$

$$+\sum_{i=0}^{n-1}\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_1\sum_{k=0}^{n-1} Q_{i,k}\left(\mathbf{A}^k\right)^{\mathrm{T}}\mathbf{c}\Bigg] \tag{3.47d}$$

$$= \frac{1}{2}\Bigg[-\sum_{i=0}^{n-1}\left(y_d^{(i)}(T) - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_1\right)\sum_{k=0}^{n-1} Q_{i,k}\left(\mathbf{A}^k\right)^{\mathrm{T}}\mathbf{c} \tag{3.47e}$$

$$-\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)}(T) - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_1\right)\Bigg]\ . \tag{3.47f}$$

Since $Q_{i,k} = Q_{k,i}$ (see, Theorem 3.1), it finally follows that

$$\left(\frac{\partial \varphi}{\partial \mathbf{x}_1}\right)^{\mathrm{T}}(\mathbf{x}_1) = \boldsymbol{\lambda}(T) = \sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\right)\Bigg|_{t=T}\ . \tag{3.48}$$

The last step now is to insert (3.40) into (3.39c), which leads to

$$u = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x} + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\right)\right)\ , \tag{3.49}$$

since

$$\begin{aligned}&\frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})^2}\underbrace{\mathbf{b}^{\mathrm{T}}\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}}_{\overset{(3.24)}{=}\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{n-1})^{\mathrm{T}}\mathbf{c}}\sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\right)\\ &= \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\sum_{k=0}^{n-1} Q_{n-1,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}\right)\ .\end{aligned} \tag{3.50}$$

□

From this point, the nature of the chosen Ansatz function (3.33) and the optimization problem can be seen. Tacking in mind Bellman's principle of optimality [14],

An optimal policy has the property that no matter what the previous decision (i.e., controls) has been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.

it becomes clear what the solution of the co-state differential equation means. Therefore, the final cost term must be chosen in such a way that the optimal policy can reach that cost. Furthermore, it states clearly why the Riccati differential equation must be static. Therefore, constructing the optimization problem in the way proposed in (3.15) has the advantage that it eliminates the need to solve a Riccati differential equation. This allows solving the optimization problem analytically, bypassing the Riccati differential equation, simplifies the solution process, reduces computational complexity, and provides a direct analytical path for solving the optimization problem.

The following examples show the principle of how the coefficients $Q_{i,k}$ from Theorem 3.3 are calculated.

Example 3.1

The following example calculates the coefficients $Q_{i,k}$ for a linear flat SISO-system with $n = 1$. Since the dimension of the system is $n = 1$, only the condition (3.46a) has to be used, which leads to

$$Q_{0,0}Q_{0,0} = Q_0 \ , \tag{3.51}$$

and therefore

$$Q_{0,0} = \sqrt{Q_0} \ . \tag{3.52}$$

Inserting the solution into the control law (3.34) and applying it to the system leads to the exponential stable error dynamics

$$\dot{e} = -\sqrt{Q_0}e \tag{3.53}$$

Example 3.2

The following example calculates the coefficients $Q_{i,k}$ for a linear flat SISO-system with $n = 2$. Due to the dimension of the system $n = 2$, all three conditions (3.46) have to be used, from which it follows that

$$\begin{aligned} Q_0 - Q_{0,1}Q_{1,0} &= 0 && \text{(3.54a)} \\ Q_{0,0} - Q_{0,1}Q_{1,1} &= 0 && \text{(3.54b)} \\ Q_1 + Q_{1,0} + Q_{0,1} - Q_{1,1}^2 &= 0 && \text{(3.54c)} \\ Q_{0,0} - Q_{1,0}Q_{1,1} &= 0 \ , && \text{(3.54d)} \end{aligned}$$

must hold. By subtracting (3.54d) from (3.54b) it follows that $Q_{0,1} = Q_{1,0}$. Applying this result to (3.54a) leads to $Q_{0,1} = Q_{1,0} = \sqrt{Q_0}$. From the third equation, it follows that $Q_{1,1} = \sqrt{Q_1 + 2\sqrt{Q_0}}$ and from equation (3.54d) that $Q_{0,0} = \sqrt{Q_0}\sqrt{Q_1 + 2\sqrt{Q_0}}$. Inserting the solution into the control law (3.34) and applying it to the system leads to the exponential stable error dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\sqrt{Q_0} & -\sqrt{Q_1 + 2\sqrt{Q_0}} \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \ . \tag{3.55}$$

3.2.2 LQT-based ILC for linear flat systems

By taking in mind that the inversion-based ILC law from (2.15) which reads as

$$u_{j+1} = u_j + Le_j \ ,$$

an intuitive guess, by the use of Theorem 3.3, would be

$$u_{j+1} = u_j + \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n}(\mathbf{x}_{j+1} - \mathbf{x}_j) + \sum_{k=0}^{n-1} Q_{n-1,k} \underbrace{\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k}\mathbf{x}_{j+1}\right)}_{=\bar{e}_{j+1}^{(k)}}\right) \tag{3.56}$$

This should motivate the following problem. From Theorem 3.3, it follows that the flatness-based trajectory following control law is optimal in the sense of minimizing the tracking error up to order n. This idea is utilized here in order to build an ILC update law. Therefore, the dynamical optimization problem is formulated in an iterative setting as

$$\min_{u_{j+1}(\cdot)} \quad J(u_{j+1}) \tag{3.57a}$$
$$\text{s.t.} \quad \dot{\mathbf{x}}_{j+1} = \mathbf{A}\mathbf{x}_{j+1} + \mathbf{b}u_{j+1} \tag{3.57b}$$
$$y_{j+1} = \mathbf{c}^{\mathrm{T}}\mathbf{x}_{j+1} \ , \tag{3.57c}$$

with

$$\begin{aligned} J(u_{j+1}) = &\underbrace{\left(\frac{1}{2}\sum_{i=0}^{n-1} e_{j+1}^{(i)} \sum_{i=0}^{n-1} Q_{i,k} e_{j+1}^{(k)}\right)\Bigg|_{t=T}}_{=\varphi(\mathbf{x}_{j+1}(T))} \\ &+ \frac{1}{2}\int_0^T \sum_{i=0}^{n-1} e_{j+1}^{(i)} Q_i e_{j+1}^{(i)} + (e_{j+1}^{(n)} - e_j^{(n)})(e_{j+1}^{(n)} - e_j^{(n)})\,\mathrm{d}t \end{aligned} \tag{3.58}$$

and

$$e_{j+1}^{(i)} = y_{d,j+1}^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{i}\mathbf{x}_{j+1} \quad , \quad \forall i \in [0, n-1] \tag{3.59a}$$
$$e_j^{(n)} = y_{d,j}^{(n)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n}\mathbf{x}_j - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_j \tag{3.59b}$$
$$e_{j+1}^{(n)} = y_{d,j+1}^{(n)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n}\mathbf{x}_{j+1} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_{j+1} \ . \tag{3.59c}$$

Here, compared to the LQT-based ILC from 3.1, the iteration dependency is formulated slightly different to build a general framework that can be applied to nonlinear systems. Therefore, the desired trajectory $y_{d,j}^{(i)}$ is chosen iterative dependent. This has the advantage that the optimization problem does not have to be formulated in the state difference over the iterations. Solving this problem leads to the following Theorem.

Theorem 3.4 (Flatness based ILC update law for a linear system)**.** *Assume the linear SISO-system (3.1) has a relative degree $r = n$. The unique solution of the optimization problem (3.57) is given by the control law*

$$u_{j+1} = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\Big(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n}\mathbf{x}_{j+1} + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\Big(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k}\mathbf{x}_{j+1}\Big)\Big) \tag{3.60}$$

with

$$y_{d,j+1}^{(k)} = y_{d,j}^{(k)} + y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k}\mathbf{x}_j \quad , \quad \forall k \in [0, n-1] \tag{3.61}$$

and the solution of the co-state differential equations

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\Big(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{i}\Big)^{\mathrm{T}}\sum_{k=0}^{n-1} Q_{i,k}\Big(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k}\mathbf{x}\Big) \ , \tag{3.62}$$

iff

$$Q_i + 2Q_{i,i-1} - Q_{i,n-1}Q_{n-1,i} = 0 \ , \quad \forall i \in [0, n-1] \tag{3.63a}$$

$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} = 0 \ , \quad \forall i,k \in [0, n-1] \ \text{and} \ i \neq k \tag{3.63b}$$

$$Q_{i,k} = Q_{k,i} \ , \tag{3.63c}$$

with $Q_i > 0$.

Proof. For the optimization problem (3.57), the Hamiltonian function follows as

$$\begin{aligned} H = \frac{1}{2}\Bigg[&\sum_{i=0}^{n-1} y_{d,j+1}^{(i)}Q_i y_{d,j+1}^{(i)} - 2\mathbf{x}_{j+1}^{\mathrm{T}}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i y_{d,j+1}^{(i)} + \mathbf{x}_{j+1}^{\mathrm{T}}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_{j+1} \\ &+ \Delta y_d^{(n)}\Big(\Delta y_d^{(n)} - 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_{j+1} - 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_{j+1} + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j\Big) \\ &- \mathbf{x}_{j+1}^{\mathrm{T}}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\Big(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_{j+1} - 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_{j+1} + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j\Big) \\ &- u_{j+1}^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{n-1})^{\mathrm{T}}\mathbf{c}\Big(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_{j+1} + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j\Big) \\ &+ \mathbf{x}_j^{\mathrm{T}}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\Big(\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + 2\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j\Big) \\ &+ u_j^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}(\mathbf{A}^{n-1})^{\mathrm{T}}\mathbf{c}\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j\Bigg] \\ &+ \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{A}\mathbf{x}_{j+1} + \mathbf{b}u_{j+1}) \ , \end{aligned} \tag{3.64}$$

with $\Delta y_d^{(n)} = y_{d,j+1}^{(n)} - y_{d,j}^{(n)}$. By applying the optimality conditions it follows

$$\dot{\mathbf{x}}_{j+1} = \mathbf{A}\mathbf{x}_{j+1} + \mathbf{b}u_{j+1} \tag{3.65a}$$

$$\dot{\boldsymbol{\lambda}} = \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\Big(y_{d,j+1}^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_{j+1}\Big) - \Big(\mathbf{A}^{\mathrm{T}} - \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}(\mathbf{A}^n)^{\mathrm{T}}\mathbf{c}\mathbf{b}^{\mathrm{T}}\Big)\boldsymbol{\lambda} \tag{3.65b}$$

$$u_{j+1} = -\frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})^2}\mathbf{b}^{\mathrm{T}}\boldsymbol{\lambda} + \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\Big(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n(\mathbf{x}_{j+1} - \mathbf{x}_j) + \Delta y_d^{(n)} + \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_j\Big) \ . \tag{3.65c}$$

To proceed with the analysis, an Ansatz function for the co-state differential equation, similar to the one used previously, is introduced as

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\left(\mathbf{A}^i\right)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right) . \tag{3.66}$$

Inserting the Ansatz function into (3.65b) leads to

$$\dot{\boldsymbol{\lambda}} = \sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_i\left(y_{d,j+1}^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_{j+1}\right) + \sum_{i=1}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i-1,k}\left(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right) , \tag{3.67}$$

similarly to the optimal co-state differential equation form (3.41) and (3.42). Differentiating the Ansatz function (3.66) with respect to time yields

$$\dot{\boldsymbol{\lambda}} = -\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,k}\left(y_{d,j+1}^{(k+1)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{k+1}\mathbf{x}_{j+1} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{b}u_{j+1}\right) . \tag{3.68}$$

By substituting the control input u_{j+1} from (3.65c) and using (3.66), it follows that the differential equation for the co-states derived from the Ansatz function is written as

$$\begin{aligned}\dot{\boldsymbol{\lambda}} = &-\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=1}^{n-1} Q_{i,k-1}\left(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right)\\ &+\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}\sum_{k=0}^{n-1} Q_{i,n-1}Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right)\\ &-\sum_{i=0}^{n-1}(\mathbf{A}^i)^{\mathrm{T}}\mathbf{c}Q_{i,n-1}\left(\Delta y_d^{(n)} - y_{d,j+1}^{(n)} + \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j\right) .\end{aligned} \tag{3.69}$$

By taking a closer look at the third part of (3.69), especially at $\Delta y_d^{(n)} - y_{d,j+1}^{(n)} + \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j$ and comparing it with it (3.67), it follows that

$$\Delta y_d^{(n)} - y_{d,j+1}^{(n)} + \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j + \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j \overset{!}{=} 0 \tag{3.70}$$

in order that the Ansatz function (3.66) is a solution for the optimal co-state differential equation (3.67). Enforcing that $y_{d,j+1}^{(n)} = y_d^{(n)}$ leads to

$$\Delta y_d^{(n)} = y_d^{(n)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j - \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j , \tag{3.71}$$

and therefore to

$$y_{d,j+1}^{(n)} = y_{d,j}^{(n)} + \Delta y_d^{(n)} = y_{d,j}^{(n)} + y_d^{(n)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_j - \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{b}u_j . \tag{3.72}$$

Using the flat parametrization (3.12), it follows that

$$y_{d,j+1}^{(i)} = y_{d,j}^{(i)} + \Delta y_d^{(i)} = y_{d,j}^{(i)} + y_d^{(i)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^i\mathbf{x}_j \quad , \quad \forall i \in [0, n-1] . \tag{3.73}$$

Inserting (3.72), (3.73) and the Ansatz function (3.66) into (3.65c) leads to (3.60). The coefficients $Q_{i,k}$ of the Ansatz function (3.66) and the final condition for the co-state can be derived similarly to the proof of Theorem 3.3. □

The ILC law (3.60) is indeed equivalent to the intuitive guess (3.56), i.e,

$$u_{j+1} = u_j + \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n(\mathbf{x}_{j+1} - \mathbf{x}_j) + \sum_{k=0}^{n-1} Q_{n-1,k}\underbrace{\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right)}_{=\bar{e}_{j+1}^{(k)}}\right) .$$

To show this, one has to start with the zeroth iteration, where

$$\bar{e}_0^{(k)} = y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_0 . \tag{3.74}$$

Therefore, the control law for the zeroth iteration follows as

$$u_0 = \underbrace{u_{-1}}_{=0} + \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_0 + \underbrace{\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_{-1}}_{=y_d^{(n)}} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_0\right)\right) . \tag{3.75}$$

Here, $\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_{-1} = y_d^{(n)}$ is chosen in order to get the optimal solution of the LQT problem for linear flat systems for the zeroth iteration. However, it should be noted at this point that this is not necessary for the algorithm itself and that this is an assumption to the system of the -1-st iteration. For similar reasons, u_{-1} is chosen to be zero. To proceed with the first iteration, it follows

$$u_1 = u_0 + \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n(\mathbf{x}_1 - \mathbf{x}_0) + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\underbrace{y_d^{(k)}}_{=y_{d,0}^{(k)}} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_1\right)\right) , \tag{3.76}$$

and inserting the zeroth iteration from (3.75), leads to

$$u_1 = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_1 + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\underbrace{y_{d,0}^{(k)} + y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_0}_{=y_{d,1}^{(k)}} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_1\right)\right) . \tag{3.77}$$

Repeating this schema to the $j+1$-st iteration, it finally follows

$$u_{j+1} = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_{j+1} + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right)\right) , \tag{3.78}$$

with

$$y_{d,j+1}^{(k)} = y_{d,j}^{(k)} + y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_j . \tag{3.79}$$

From this point, the nature of the ILC can be seen. The so-derived ILC algorithm summarizes the trajectory error over the iteration. In other words, it approximates the mean plant's dynamic behaviour.

3.3 Flatness-based ILC with observer

This section presents a method for the case that not all states of the system can be measured, but only a noisy measurement of the output is available. Therefore, a state observer for the not-measured states has to be used. For this case, the system is assumed in the form

$$\begin{aligned}\dot{\mathbf{x}}_{j+1} &= \mathbf{A}\mathbf{x}_{j+1} + \mathbf{b}u_{j+1} \\ y_{j+1} &= \mathbf{c}^{\mathrm{T}}\mathbf{x}_{j+1}\ ,\end{aligned} \tag{3.80}$$

where $\mathbf{x}_{j+1} \in \mathbb{R}^n$ is the state and $y_{j+1} \in \mathbb{R}$ is the output. Furthermore, the system is flat concerning the output y_{j+1}. Therefore, the system can be transformed into

$$\begin{aligned}\dot{z}_{1,j+1} &= z_{2,j+1} \\ &\vdots \\ \dot{z}_{n,j+1} &= \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{T}^{-1}\mathbf{z}_{j+1} + \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_{j+1}\end{aligned} \tag{3.81}$$

by the use of the diffeomorphism

$$\mathbf{z}_{j+1} = \underbrace{\begin{bmatrix}\mathbf{c}^{\mathrm{T}} \\ \mathbf{c}^{\mathrm{T}}\mathbf{A} \\ \vdots \\ \mathbf{c}^{\mathrm{T}}\mathbf{A}^{(n-1)}\end{bmatrix}}_{=\mathbf{T}}\mathbf{x}_{j+1}\ . \tag{3.82}$$

Introducing the error in the form

$$\mathbf{z}_{e,j+1} = \mathbf{z}_{j+1} - \underbrace{\begin{bmatrix} y_d \\ y_d^{(1)} \\ \vdots \\ y_d^{(n-1)}\end{bmatrix}}_{=\mathbf{y}_d}\ , \tag{3.83}$$

leads to the observability canonical form (*Brunovsky normal form*)

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} z_{1e,j+1} \\ z_{2e,j+1} \\ \vdots \\ z_{ne,j+1}\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0\end{bmatrix}\begin{bmatrix} z_{1e,j+1} \\ z_{2e,j+1} \\ \vdots \\ z_{ne,j+1}\end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1\end{bmatrix}\tilde{u}_{j+1} \tag{3.84}$$

with $\tilde{u}_{j+1} = \mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{T}^{-1}(\mathbf{z}_{e,j+1} + \mathbf{y}_d) + \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b}u_{j+1} - y_d^{(n)}$. From this equation, it can be seen that the error dynamics can be written as

$$\dot{\mathbf{z}}_{e,j+1} = \tilde{\mathbf{A}}\mathbf{z}_{e,j+1} + \tilde{\mathbf{b}}\tilde{u}_{j+1} + \mathbf{G}\mathbf{w}_{j+1} \tag{3.85a}$$

$$y_{e,j+1} = \tilde{\mathbf{c}}^{\mathrm{T}}\mathbf{z}_{e,j+1} + v_{j+1}\ , \tag{3.85b}$$

with an additional process noise $\mathbf{w}_{j+1} \in \mathbb{R}^n$ in order to model a stochastic model plant mismatch and the measurement noise $v_{j+1} \in \mathbb{R}$. We assume the process noise to have zero mean, i.e., $\mathbb{E}(w_{j+1}) = 0$, and the correlation $\mathbb{E}(w_{j+1}(\tau), w_{j+1}(t)) = \mathbf{Q}\delta(t-\tau)$, which is symbolized as $w_{j+1} \sim (0, \mathbf{Q})$. The measurement noise $v_{j+1} \in \mathbb{R}$ is assumed to have zero mean as well, i.e., $\mathbb{E}(v_{j+1}) = 0$, and the correlation $\mathbb{E}(v_{j+1}(\tau), v_{j+1}(t)) = R\delta(t-\tau)$, which is symbolized as $v_{j+1} \sim (0, R)$.

Remark 3. *The question arises of whether it is possible to estimate the states of the system directly. Transforming the system to the observability canonical form is necessary to construct a general framework that can also be used in the nonlinear case. Additionally, it has some advantages in the linear case as well. To apply stochastic optimal Kalman filtering, the stochastic disturbance must be Gaussian with zero mean. For this reason a mean-free system is introduced. Therefore, the physically modelled system is assumed to be the ground truth since it is the best we have. The unmodelled plant dynamic, induced by model plant mismatches, is thus assumed to be Gaussian with zero mean concerning the nominal system. This might only be true approximately since the derivative of a Gaussian distribution is not Gaussian anymore.*

The goal now is to construct a state observer for the system given in (3.85). Following [17], one can use a linear observer of the form

$$\dot{\hat{\mathbf{z}}}_{e,j+1} = \tilde{\mathbf{A}}\hat{\mathbf{z}}_{e,j+1} + \tilde{\mathbf{b}}\tilde{u}_{j+1} + \mathbf{k}^f\left(y_{e,j+1} - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{z}}_{e,j+1}\right) , \tag{3.86}$$

where the stochastic optimal choice for $\mathbf{k}^f$ is given by a Kalman filter [17], which can be written as

$$\begin{aligned} \mathbf{k}^f &= \mathbf{P}^f\mathbf{c}(R)^{-1} && (3.87\text{a})\\ \dot{\mathbf{P}}^f &= \tilde{\mathbf{A}}\mathbf{P}^f + \mathbf{P}^f\tilde{\mathbf{A}}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} - \mathbf{k}^f R\left(\mathbf{k}^f\right)^{\mathrm{T}} && (3.87\text{b})\\ \dot{\hat{\mathbf{z}}}^f_{j+1} &= \tilde{\mathbf{A}}\hat{\mathbf{z}}^f_{j+1} + \tilde{\mathbf{b}}\tilde{u}_{j+1} + \mathbf{k}^f\left(y_{j+1} - \tilde{\mathbf{c}}^{\mathrm{T}}\hat{\mathbf{z}}^f_{j+1}\right) , && (3.87\text{c}) \end{aligned}$$

with the initial conditions

$$\begin{aligned} \hat{\mathbf{z}}^f_{j+1}(0) &= \mathbf{z}_{j+1}(0) && (3.87\text{d})\\ \mathbf{P}^f(0) &= \mathbf{P}_0 \ . && (3.87\text{e}) \end{aligned}$$

From Theorem 3.4, it follows that the optimal flatness-based ILC, in order to minimize the error up to order n, with full state information, is given by

$$u_{j+1} = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{x}_{j+1} + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y^{(k)}_{d,j+1} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_{j+1}\right)\right) , \tag{3.88}$$

and the trajectory update law as

$$y^{(k)}_{d,j+1} = y^{(k)}_{d,j} + y^{(k)}_d - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{x}_j \quad , \quad \forall k \in [0, n-1] \ . \tag{3.89}$$

Since the trajectory update law (3.89) is calculated offline (after every iteration), it only contains quantities from the last iteration j. Therefore, a Kalman smoother is used, which can be written as [17]

$$\mathbf{K}^s = \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}}\left(\mathbf{P}^f\right)^{-1} \tag{3.90a}$$

$$\dot{\mathbf{P}}^s = \left(\tilde{\mathbf{A}} + \mathbf{K}^s\right)\mathbf{P}^s + \mathbf{P}^f\left(\tilde{\mathbf{A}} + \mathbf{K}^s\right)^{\mathrm{T}} - \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} \tag{3.90b}$$

$$\dot{\hat{\mathbf{z}}}^s_{j+1} = \tilde{\mathbf{A}}\hat{\mathbf{z}}^s_{j+1} + \mathbf{K}^s\left(\hat{\mathbf{z}}^s_{j+1} - \hat{\mathbf{z}}^f_{j+1}\right) , \tag{3.90c}$$

with the final conditions

$$\hat{\mathbf{z}}^s_{j+1}(T) = \hat{\mathbf{z}}^f_{j+1}(T) \tag{3.90d}$$

$$\mathbf{P}^s(T) = \mathbf{P}^f(T) \tag{3.90e}$$

The superscript $(\cdot)^f$ denotes the filter or forward-pass, and $(\cdot)^s$ is the smoother. Since the flatness-based ILC algorithm from Section 3.2 consists of an online (feedback) and an offline phase, the estimations of the Kalman filter are used for the feedback part and the estimations of the Kalman smoother for the trajectory update. Therefore, rewriting (3.88) as

$$u_{j+1} = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\left(-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\hat{\mathbf{x}}^f_{j+1} + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\hat{y}^{(k),s}_{d,j+1} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\hat{\mathbf{x}}^f_{j+1}\right)\right) \tag{3.91}$$

and keeping in mind that

$$\hat{z}^f_{ie} = \hat{y}^{(i-1),f}_{j+1} - y_d^{(i-1)} = \mathbf{c}^{\mathrm{T}}\mathbf{A}^{i-1}\hat{\mathbf{x}}^f_{j+1} - y_d^{(i-1)} , \tag{3.92}$$

the observed error state can be written as

$$\hat{\mathbf{z}}^f_{e,j+1} = \underbrace{\begin{bmatrix} \mathbf{c}^{\mathrm{T}} \\ \mathbf{c}^{\mathrm{T}}\mathbf{A} \\ \vdots \\ \mathbf{c}^{\mathrm{T}}\mathbf{A}^{(n-1)} \end{bmatrix}}_{=\mathbf{T}} \hat{\mathbf{x}}^f_{j+1} - \underbrace{\begin{bmatrix} y_d \\ y_d^{(1)} \\ \vdots \\ y_d^{(n-1)} \end{bmatrix}}_{=\mathbf{y}_d} . \tag{3.93}$$

From this point, it follows that the state can be written as

$$\hat{\mathbf{x}}^f_{j+1} = \mathbf{T}^{-1}\left(\hat{\mathbf{z}}^f_{e,j+1} + \mathbf{y}_d\right) , \tag{3.94}$$

and therefore (3.91) follows as

$$\begin{aligned} u_{j+1} = \frac{1}{(\mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1}\mathbf{b})}\Bigg(&-\mathbf{c}^{\mathrm{T}}\mathbf{A}^n\mathbf{T}^{-1}\left(\hat{\mathbf{z}}^f_{e,j+1} + \mathbf{y}_d\right) + y_d^{(n)} \\ &+ \sum_{k=0}^{n-1} Q_{n-1,k}\left(\hat{y}^{(k),s}_{d,j+1} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{T}^{-1}\left(\hat{\mathbf{z}}^f_{e,j+1} + \mathbf{y}_d\right)\right)\Bigg) . \end{aligned} \tag{3.95}$$

For the trajectory update, the Kalman smoother is applied to the error dynamics and therefore

$$\hat{\mathbf{x}}_j^s = \mathbf{T}^{-1}\Big(\hat{\mathbf{z}}_{e,j}^s + \mathbf{y}_d\Big) \,. \tag{3.96}$$

From (3.89) it follows that

$$\hat{y}_{d,j+1}^{(k),s} = \hat{y}_{d,j}^{(k),s} + y_d^{(k)} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^k\mathbf{T}^{-1}\Big(\hat{\mathbf{z}}_{e,j}^s + \mathbf{y}_d\Big)\,, \qquad \forall i \in [0, n-1] \,, \tag{3.97}$$

which can be written in vectorized form as

$$\hat{\mathbf{y}}_{d,j+1}^s = \hat{\mathbf{y}}_{d,j}^s - \hat{\mathbf{z}}_{e,j}^s \,. \tag{3.98}$$

The ILC-update law (3.95) and the trajectory update law (3.98), which is calculated in the offline phase (after every iteration), build up the centrepiece of the flatness-based ILC with an error observer. However, ILC laws also need to deal with repetitive disturbances. Therefore, the process noise $\mathbf{w}_{j+1}$, in general, is no longer white Gaussian noise anymore but a coloured process noise that can be incorporated in this approach by filtering white noise, see, e.g., [17].

Remark 4. *Keeping in mind that the control law (3.95) is equivalent to the intuitive guess (3.56), it follows that the Kalman smoother in the iteration-dependent trajectory is related to the learning filter* $\mathbf{L}$ *from the discrete-time case.*

3.4 Example

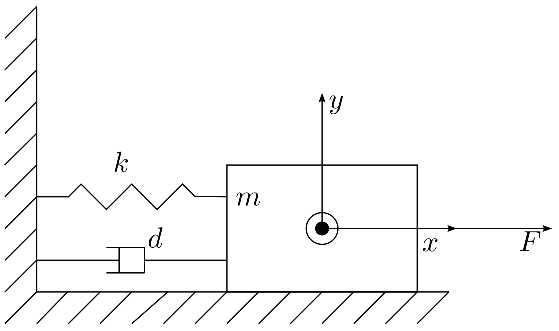
This section shows the results presented in this chapter on the example of the spring-mass-damper system given in Figure 3.1 with the mass $m = 1\,\mathrm{kg}$, the damping coefficient $d = 1\,\mathrm{kg/s}$, the spring stiffness $k = 1\,\mathrm{kg/s^2}$, and the external force F. By applying Newton's second law, the state space representation follows as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F_d \,, \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{3.99a}$$

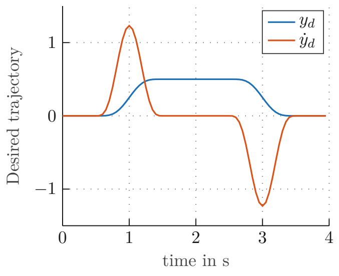
and the output as

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \,. \tag{3.100a}$$

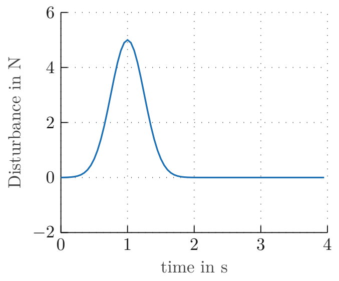
Here, the input u of the system is given as a force F, and F_d is the external disturbance, acting on the center of gravity (cog). The system is flat concerning the output of the system y, which represents the position of the cog with respect to the given coordinate system. The desired trajectory for each iteration can be seen in Figure 3.1b and, the disturbance F_d, which is activated after 10 iterations acting in the same direction as F, can be seen in Figure 3.1c. Figure 3.2 shows the L_2 norm of the tracking error for the presented ILC strategies. In Figure 3.2a, it can be seen that the LQT-based ILC has the ability to suppress the external disturbance, which is applied to the system after the 10-th iteration. As it could be seen in Figure 3.2b, the tracking error of the flatness-based



(a) Schematic representation of a spring mass damper system.



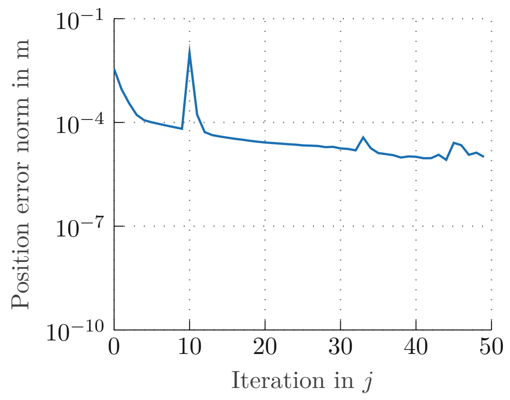
(b) Desired trajectory.



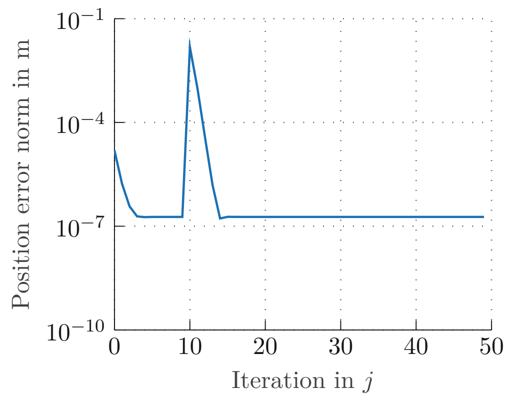
(c) External disturbance force $F_d(t)$.

Figure 3.1: Schematic representation, desired trajectory, and applied disturbance.

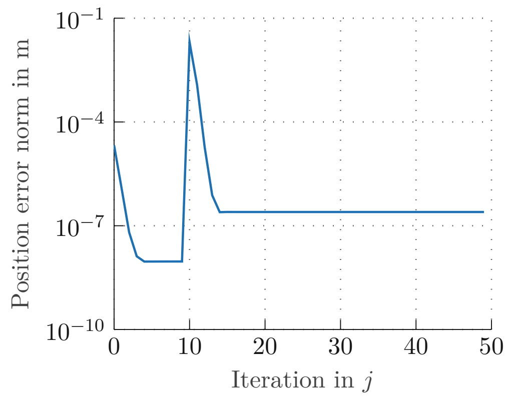
ILC algorithm converges way faster than the LQT-based ILC. Figure 3.2c shows the flatness-based ILC with an error observer. Here, it can be seen that in the first iterations, before the disturbance is applied, the norm of the error decreases to a value smaller than that of the flatness-based ILC. This could be due to numerical errors that occur due to the discrete nature of the simulation. These errors could be reduced since the trajectory update has the potential to minimize errors due to the Kalman smoother's ability to estimate states without introducing phase shifts. After the disturbance is applied to the system, the flatness-based ILC with error observer does not converge to the same value as before applying the disturbance. This is due to the unmodelled disturbance in the observer since the disturbance is modelled as a stochastic disturbance with zero mean. Therefore, to get rid of this behaviour, one could extend the error observer by a disturbance observer and a first-order Markov process [14] as described in Section 3.3, which leads to the convergence behaviour in Figure 3.2d. The corresponding control inputs are depicted in Figure 3.3 to show that these comparisons are fair. Here, it can be seen that they look qualitatively similar for the 0-th, 10-th, and 50-th iteration.



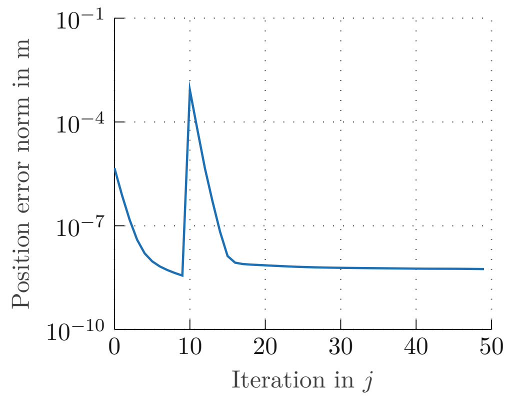
(a) LQT-based ILC with full state information.



(b) Flatness-based ILC with full state information.

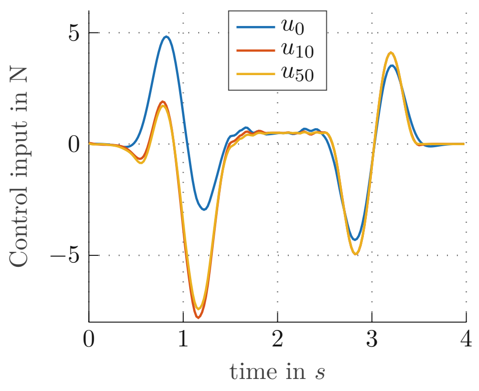


(c) Flatness-based ILC with error dynamics observer.

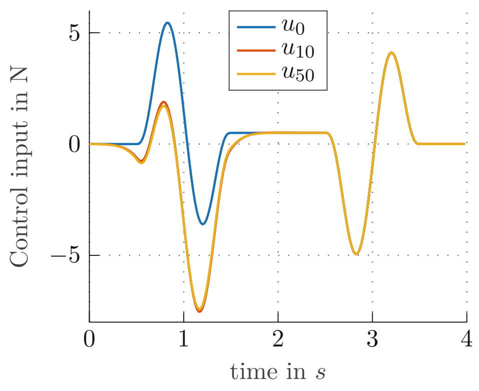


(d) Flatness-based ILC with error dynamics and disturbance observer.

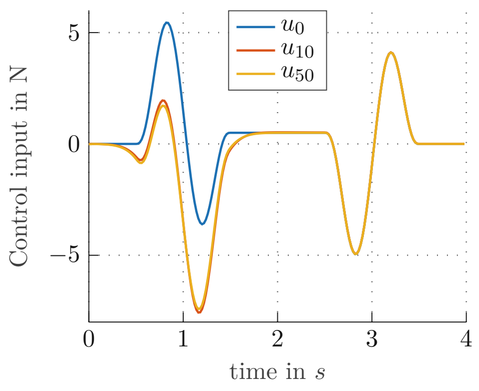
Figure 3.2: Comparison of the convergence behaviour of different ILC algorithms.



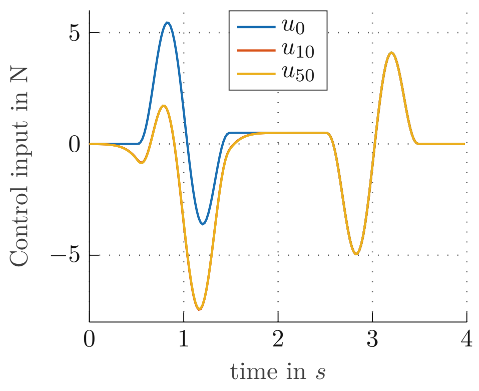
(a) LQT-based ILC with full state information.



(b) Flatness-based ILC with full state information.



(c) Flatness-based ILC with error dynamics observer.



(d) Flatness-based ILC with error dynamics and disturbance observer.

Figure 3.3: Comparison of the control input of different ILC algorithms.

4 ILC for nonlinear AI-systems

This chapter considers a single input, single output (SISO) nonlinear affine input system (AI-system) of the form

$$\dot{\mathbf{x}}_j(t) = \mathbf{f}(\mathbf{x}_j(t)) + \mathbf{g}(\mathbf{x}_j(t))u_j(t)\,, \qquad \mathbf{x}_j(0) = \mathbf{x}_0 \tag{4.1a}$$
$$y_j(t) = h(\mathbf{x}_j(t))\ , \tag{4.1b}$$

where $\mathbf{x}_j(t)$ denotes the state for the $j-th$ iteration with the initial condition $\mathbf{x}_j(0) = \mathbf{x}_0$, the control input $u_j(t)$, and the output $y_j(t)$ for all $j \in \mathbb{N}$. Here $\mathbf{f}(\mathbf{x}_j(t)) \in \mathbb{R}^n$ and $\mathbf{g}(\mathbf{x}_j(t)) \in \mathbb{R}^n$ are both smooth vector fields and $h(\mathbf{x}_j(t)) \in \mathbb{R}$ is a smooth function.

Remark 5. *This chapter focuses exclusively on the SISO (single input, single output) case to maintain notational simplicity. This choice enhances clarity, facilitates a solid foundation for understanding core concepts, reduces mathematical complexity, and allows easier visualization. Still, the ILC strategies presented in this chapter are not restricted to the SISO case and can be extended to the MIMO case. For the same reason, the time argument t is omitted when the context implies its presence.*

4.1 LQT-based ILC

This section presents how the LQT-based ILC from Section 3.1 can be extended to systems of the form (4.1). Therefore, the system is linearized around a given trajectory $(\tilde{\mathbf{x}}, \tilde{u})$. For this reason, the system variables and the output variable are expressed as small perturbations around this trajectory as

$$\mathbf{x}_j = \tilde{\mathbf{x}} + \Delta\mathbf{x}_j \tag{4.2a}$$
$$u_j = \tilde{u} + \Delta u_j \tag{4.2b}$$
$$y_j = \tilde{y} + \Delta y_j\ . \tag{4.2c}$$

Inserting (4.2) into (4.1) leads to

$$\dot{\tilde{\mathbf{x}}} + \Delta\dot{\mathbf{x}}_j = \mathbf{f}(\tilde{\mathbf{x}} + \Delta\mathbf{x}_j) + \mathbf{g}(\tilde{\mathbf{x}} + \Delta\mathbf{x}_j)(\tilde{u} + \Delta u)\,, \qquad \mathbf{x}_j(0) = \tilde{\mathbf{x}}_0 + \Delta\mathbf{x}_{j,0} \tag{4.3a}$$
$$\tilde{y} + \Delta y_j = h(\tilde{\mathbf{x}} + \Delta\mathbf{x}_j)\ . \tag{4.3b}$$

Applying the second-order Taylor formula and neglecting higher-order terms yields

$$\dot{\tilde{\mathbf{x}}} + \Delta\dot{\mathbf{x}}_j = \underbrace{\mathbf{f}(\tilde{\mathbf{x}}) + \mathbf{g}(\tilde{\mathbf{x}})\tilde{u}}_{=\dot{\tilde{\mathbf{x}}}} + \left(\frac{\partial\mathbf{f}}{\partial\mathbf{x}}(\tilde{\mathbf{x}}) + \frac{\partial\mathbf{g}}{\partial\mathbf{x}}(\tilde{\mathbf{x}})\tilde{u}\right)\Delta\mathbf{x}_j + \mathbf{g}(\tilde{\mathbf{x}})\Delta u_j \tag{4.4a}$$

$$\tilde{y} + \Delta y_j = \underbrace{h(\tilde{\mathbf{x}})}_{=\tilde{y}} + \frac{\partial h}{\partial\mathbf{x}}(\tilde{\mathbf{x}})\Delta\mathbf{x}_j \tag{4.4b}$$

and the linearization of the system around a trajectory $(\tilde{\mathbf{x}}, \tilde{u})$ can be written as

$$\Delta\dot{\mathbf{x}}_j = \mathbf{A}(t)\Delta\mathbf{x}_j + \mathbf{b}(t)\Delta u_j \,, \qquad \Delta\mathbf{x}_j(0) = \Delta\mathbf{x}_{j,0} = \mathbf{x}_{j,0} - \tilde{\mathbf{x}}_0 \tag{4.5a}$$

$$\Delta y_j = \mathbf{c}^{\mathrm{T}}(t)\Delta\mathbf{x}_j \tag{4.5b}$$

with

$$\mathbf{A}(t) = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}(\tilde{\mathbf{x}}) + \frac{\partial\mathbf{g}}{\partial\mathbf{x}}(\tilde{\mathbf{x}})\tilde{u} \,, \qquad \mathbf{b}(t) = \mathbf{g}(\tilde{\mathbf{x}}) \,, \tag{4.5c}$$

$$\mathbf{c}^{\mathrm{T}}(t) = \frac{\partial h}{\partial\mathbf{x}}(\tilde{\mathbf{x}}) \,. \tag{4.5d}$$

For the sake of readability, time dependencies are left away in further investigations. Let us assume that $(\mathbf{x}_d, u_d)$ is a trajectory of the system (4.1) for the desired flat output y_d. Using the linearization around this given trajectory, the iteration-dependent error system yields

$$\underbrace{\Delta\dot{\mathbf{x}}_{j+1} - \Delta\dot{\mathbf{x}}_j}_{=\dot{\mathbf{z}}_{j+1}} = \mathbf{A}\left(\underbrace{\Delta\mathbf{x}_{j+1} - \Delta\mathbf{x}_j}_{=\mathbf{z}_{j+1}}\right) + \mathbf{b}\left(\underbrace{\Delta u_{j+1} - \Delta u_j}_{v_{j+1}}\right), \qquad \mathbf{z}_{j+1} = \mathbf{x}_{j+1,0} - \mathbf{x}_{j,0} \,, \tag{4.6}$$

with the tracking error

$$e_{j+1} = y_d - \underbrace{(y_d + \Delta y_{j+1})}_{=y_{j+1}} = -\Delta y_{j+1} \,. \tag{4.7}$$

The optimization problem for the LQT-based ILC, with the positive weights F, Q and R, can be written as

$$\min_{v_{j+1}(\cdot)} J(v_{j+1}) = \frac{1}{2}(e_{j+1}Fe_{j+1})\big|_{t=T} + \frac{1}{2}\int_0^T e_{j+1}Qe_{j+1} + v_{j+1}Rv_{j+1}\,\mathrm{d}t \tag{4.8a}$$

$$\text{s.t.} \quad \dot{\mathbf{z}}_{j+1} = \mathbf{A}\mathbf{z}_{j+1} + \mathbf{b}v_{j+1} \tag{4.8b}$$

$$e_{j+1} = -\Delta y_{j+1} = \mathbf{c}^{\mathrm{T}}\Delta\mathbf{x}_j - \mathbf{c}^{\mathrm{T}}\Delta\mathbf{x}_j - \mathbf{c}^{\mathrm{T}}\Delta\mathbf{x}_{j+1} \tag{4.8c}$$

$$= e_j - \mathbf{c}^{\mathrm{T}}\mathbf{z}_{j+1} \tag{4.8d}$$

where $\mathbf{z}_{j+1} = \Delta\mathbf{x}_{j+1} - \Delta\mathbf{x}_j$ describes the difference over the iteration of the state and $v_{j+1} = \Delta u_{j+1} - \Delta u_j$ the difference in the control input. Solving this optimization problem is fully equivalent to (3.2), except that the iteration-dependent error system is *time-variant*. Hence, to utilize the algorithm of the LQT-based ILC for nonlinear AI-systems, (4.1) has to be linearized around a given trajectory $(\tilde{\mathbf{x}}_d, \tilde{u}_d)$. By the use of this linearization, (3.7) has to be solved backwards in time with the corresponding terminal conditions (3.8). In this formulation, the feedforward term implies the knowledge of the tracking error e_j from the previous iteration. The solution of the Riccati differential equation $\mathbf{S}$ is fixed over the iterations and the external forcing term $\mathbf{g}$ has to be recalculated after every iteration. For the zeroth iteration, $\Delta\mathbf{x}_{-1}$, Δu_{-1}, and Δy_{-1} are set to zero, and therefore, the ILC update law implies the classical LQT formulation of [13, 14] applied to nonlinear AI-systems.

In order to construct the iteration-dependent error system, the system matrices have to be identical over the iterations. In other words, the system must be linearized around the same trajectory $(\tilde{\mathbf{x}}_d, \tilde{u}_d)$ for each iteration j to apply the error evolution over the iterations. Therefore, the question arises if it is possible to improve the performance of LQT-based ILC by not simply linearizing the system around an initial trajectory for each iteration j, but by linearizing around the trajectory $(\tilde{\mathbf{x}}_j, \tilde{u}_j)$ of the last iteration. For this reason, the system variables and the output variable are expressed as small perturbations around this trajectory as

$$\mathbf{x}_{j+1} = \tilde{\mathbf{x}}_{j+1} + \Delta\mathbf{x}_{j+1} \tag{4.9a}$$
$$u_{j+1} = \tilde{u}_{j+1} + \Delta u_{j+1} \tag{4.9b}$$
$$y_{j+1} = \tilde{y}_{j+1} + \Delta y_{j+1} \ . \tag{4.9c}$$

Applying a linearization of the system around the trajectory $(\tilde{\mathbf{x}}_{j+1}, \tilde{u}_{j+1})$, the linearized system follows as

$$\Delta\dot{\mathbf{x}}_{j+1} = \underbrace{\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_{j+1}) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_{j+1})\tilde{u}_{j+1}\right)}_{=\mathbf{A}_{j+1}} \Delta\mathbf{x}_{j+1} + \underbrace{\mathbf{g}(\tilde{\mathbf{x}}_{j+1})}_{=\mathbf{b}_{j+1}} \Delta u_{j+1} \tag{4.10a}$$
$$\Delta y_{j+1} = \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_{j+1})}_{=\mathbf{c}^{\mathrm{T}}_{j+1}} \Delta\mathbf{x}_{j+1} \ . \tag{4.10b}$$

Similar to the linearization around a fixed trajectory, one has to construct an iteration-dependent error system that yields

$$\begin{aligned} \Delta\dot{\mathbf{x}}_{j+1} - \Delta\dot{\mathbf{x}}_j &= \mathbf{A}_{j+1}\Delta\mathbf{x}_{j+1} - \mathbf{A}_j\Delta\mathbf{x}_j + \mathbf{b}_{j+1}\Delta u_{j+1} - \mathbf{b}_j\Delta u_j \ , \\ \Delta\mathbf{x}_{j+1,0} - \Delta\mathbf{x}_{j,0} &= \mathbf{x}_{j+1,0} - \mathbf{x}_{j,0} \ . \end{aligned} \tag{4.11}$$

In comparison to (4.6), the linearized states $\Delta\mathbf{x}_{j+1}$ and $\Delta\mathbf{x}_j$ could not be factorized in order to construct a linear iteration-dependent error system, since the system matrices are iteration dependent. To construct an iteration-dependent error system, the system is linearized again. Therefore, the system variables and the output variable are written as

$$\Delta\mathbf{x}_{j+1} = \Delta\mathbf{x}_j + \Delta\mathbf{z}_{j+1} \tag{4.12a}$$
$$\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \Delta\mathbf{z}_{j+1} \tag{4.12b}$$
$$\Delta u_{j+1} = \Delta u_j + \Delta v_{j+1} \tag{4.12c}$$
$$\tilde{u}_{j+1} = \tilde{u}_j + \Delta v_{j+1} \tag{4.12d}$$
$$\Delta y_{j+1} = \Delta y_j + \Delta\varepsilon_{j+1} \tag{4.12e}$$
$$\tilde{y}_{j+1} = \tilde{y}_j + \Delta\varepsilon_{j+1} \ , \tag{4.12f}$$

where $\Delta\mathbf{z}_{j+1}$ describes a small variation over the iteration for the linearized state, Δv_{j+1} for the control input, and $\Delta\varepsilon_{j+1}$ for the output. Inserting (4.12) into (4.10) leads to

$$\Delta\dot{\mathbf{x}}_j + \Delta\dot{\mathbf{z}}_{j+1} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j + \Delta\mathbf{z}_{j+1}) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j + \Delta\mathbf{z}_{j+1})(\tilde{u}_j + \Delta v_{j+1})\right)(\Delta\mathbf{x}_j + \Delta\mathbf{z}_{j+1}) \tag{4.13a}$$

$$+ \mathbf{g}(\tilde{\mathbf{x}}_j + \Delta\mathbf{z}_{j+1})(\Delta u_j + \Delta v_{j+1}) \tag{4.13b}$$

$$\Delta y_j + \Delta\varepsilon_{j+1} = \frac{\partial h}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j + \Delta\mathbf{z}_{j+1})(\Delta\mathbf{x}_j + \Delta\mathbf{z}_{j+1}) \; . \tag{4.13c}$$

Applying the second-order Taylor formula to (4.13), it follows that

$$\Delta\dot{\mathbf{x}}_j + \Delta\dot{\mathbf{z}}_{j+1} = \underbrace{\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j)\tilde{u}_j\right)\Delta\mathbf{x}_j + \mathbf{g}(\tilde{\mathbf{x}}_j)\Delta u_j}_{\Delta\dot{\mathbf{x}}_j} \tag{4.14a}$$

$$+ \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j)\tilde{u}_j\right)\Delta\mathbf{z}_{j+1} + \mathbf{g}(\tilde{\mathbf{x}}_j)\Delta v_{j+1} \tag{4.14b}$$

$$+ \Delta\mathbf{x}_j^{\mathrm{T}}\left(\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}(\tilde{\mathbf{x}}_j) + \frac{\partial^2 \mathbf{g}}{\partial \mathbf{x}^2}(\tilde{\mathbf{x}}_j)\tilde{u}_j\right)\Delta\mathbf{z}_{j+1} \tag{4.14c}$$

$$+ \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j)(\Delta\mathbf{x}_j\Delta v_{j+1} + \Delta\mathbf{z}_{j+1}\Delta u_j) \tag{4.14d}$$

$$\Delta y_j + \Delta\varepsilon_{j+1} = \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j)\Delta\mathbf{x}_j}_{\Delta y_j} + \Delta\mathbf{x}_j^{\mathrm{T}}\frac{\partial^2 h}{\partial \mathbf{x}^2}(\tilde{\mathbf{x}}_j)\Delta\mathbf{z}_{j+1} + \frac{\partial h}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j)\Delta\mathbf{z}_{j+1} \; . \tag{4.14e}$$

Assuming the second-order terms to be small, the Taylor approximation (4.14) can be written as

$$\Delta\dot{\mathbf{z}}_{j+1} = \mathbf{A}_j\Delta\mathbf{z}_{j+1} + \mathbf{b}_j\Delta v_{j+1} \, , \qquad \Delta\mathbf{z}_{j+1}(0) = \Delta\mathbf{z}_{j+1,0} = \frac{1}{2}(\mathbf{x}_{j+1,0} - \mathbf{x}_{j,0}) \tag{4.15a}$$

$$\Delta\varepsilon_{j+1} = \mathbf{c}_j^{\mathrm{T}}\Delta\mathbf{z}_{j+1} \tag{4.15b}$$

with

$$\mathbf{A}_j = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j)\tilde{u}_j \, , \qquad \mathbf{b}_j = \mathbf{g}(\tilde{\mathbf{x}}_j) \tag{4.15c}$$

$$\mathbf{c}_j^{\mathrm{T}} = \frac{\partial h}{\partial \mathbf{x}}(\tilde{\mathbf{x}}_j) \; , \tag{4.15d}$$

with the tracking error

$$e_{j+1} = y_d - y_{j+1} = y_d - \left(\underbrace{\tilde{y}_{j+1}}_{\approx y_d} + \Delta y_{j+1}\right) = -\Delta y_{j+1} \; , \tag{4.16}$$

Therefore, the optimization problem for the LQT-based ILC with trajectory evolution, and the positive weights F, Q and R, can be written as

$$\begin{aligned} \min_{\Delta v_{j+1}(\cdot)} J(\Delta v_{j+1}) &= \frac{1}{2}(e_{j+1}Fe_{j+1})\big|_{t=T} + \frac{1}{2}\int_0^T e_{j+1}Qe_{j+1} + \Delta v_{j+1}R\Delta v_{j+1}\,\mathrm{d}t && (4.17\text{a}) \\ \text{s.t.}\quad \Delta\dot{\mathbf{z}}_{j+1} &= \mathbf{A}_j\Delta\mathbf{z}_{j+1} + \mathbf{b}_j\Delta v_{j+1} && (4.17\text{b}) \\ e_{j+1} &= -\Delta y_{j+1} = -\Delta y_j - \Delta\varepsilon_{j+1} && (4.17\text{c}) \\ &= e_j - \mathbf{c}_j^{\mathrm{T}}\Delta\mathbf{z}_{j+1} && (4.17\text{d}) \end{aligned}$$

where $\Delta\mathbf{z}_{j+1} = \Delta\mathbf{x}_{j+1} - \Delta\mathbf{x}_j$ describes the difference over the iteration of the state and $\Delta v_{j+1} = \Delta u_{j+1} - \Delta u_j$ the difference of the control input. The Hamiltonian for LQT-based ILC with trajectory evolution is given by

$$\begin{aligned} H = &\frac{1}{2}(e_jQe_j - 2\Delta\mathbf{z}_{j+1}^{\mathrm{T}}\mathbf{c}Qe_j + \Delta\mathbf{z}_{j+1}^{\mathrm{T}}\mathbf{c}_jQ\mathbf{c}_j^{\mathrm{T}}\Delta\mathbf{z}_{j+1} + \Delta v_{j+1}R\Delta v_{j+1}) \\ &+ \boldsymbol{\lambda}_j^{\mathrm{T}}(\mathbf{A}_j\Delta\mathbf{z}_{j+1} + \mathbf{b}_j\Delta v_{j+1}) \end{aligned} \quad (4.18)$$

Using the optimality conditions, it follows that

$$\frac{\partial H}{\partial \Delta v}(\Delta v_{j+1}^*) = R\Delta v_{j+1}^* + (\mathbf{b}^*)_j^{\mathrm{T}}\boldsymbol{\lambda}_j^* \overset{!}{=} 0 \quad (4.19\text{a})$$

$$\left(\frac{\partial H}{\partial \Delta\mathbf{z}}(\Delta\mathbf{z}_{j+1}^*)\right)^{\mathrm{T}} = \mathbf{c}_j^*Q(\mathbf{c}^*)_j^{\mathrm{T}}\Delta\mathbf{z}_{j+1}^* - \mathbf{c}_j^*Qe_j^* + (\mathbf{A}^*)_j^{\mathrm{T}}\boldsymbol{\lambda}^* \overset{!}{=} -\dot{\boldsymbol{\lambda}}_j^*\ . \quad (4.19\text{b})$$

For readability, the star (*) for the optimal solution is dropped in the following. In order to construct differential equations for the co-state that are independent of $\Delta\mathbf{z}_{j+1}$, the standard Ansatz function [14] for the co-state

$$\boldsymbol{\lambda}_j = \mathbf{S}_j\Delta\mathbf{z}_{j+1} + \mathbf{g}_j \quad (4.20)$$

is introduced. By differentiation, the Ansatz function concerning time and setting it equal to (4.19b), it follows that

$$\dot{\boldsymbol{\lambda}}_j = \dot{\mathbf{S}}_j\Delta\mathbf{z}_{j+1} + \mathbf{S}_j\Delta\dot{\mathbf{z}}_{j+1} + \dot{\mathbf{g}}_j \overset{!}{=} -\mathbf{c}_jQ\mathbf{c}_j^{\mathrm{T}}\Delta\mathbf{z}_{j+1} + \mathbf{c}_jQe_j - \mathbf{A}_j^{\mathrm{T}}\mathbf{S}_j\Delta\mathbf{z}_{j+1} - \mathbf{A}_j^{\mathrm{T}}\mathbf{g}_j\ . \quad (4.21)$$

This leads to the well-known Riccati differential equation for $\mathbf{S}_j$ and the feedforward differential equation for the external forcing term $\mathbf{g}_j$ in the form

$$\begin{aligned} \dot{\mathbf{S}}_j &= -\mathbf{S}_j\mathbf{A}_j - \mathbf{A}_j^{\mathrm{T}}\mathbf{S}_j + \mathbf{S}_j\mathbf{b}_jR^{-1}\mathbf{b}_j^{\mathrm{T}}\mathbf{S}_j - \mathbf{c}_jQ\mathbf{c}_j^{\mathrm{T}} && (4.22\text{a}) \\ \dot{\mathbf{g}}_j &= (\mathbf{S}_j\mathbf{b}_jR^{-1}\mathbf{b}_j^{\mathrm{T}} - \mathbf{A}_j^{\mathrm{T}})\mathbf{g}_j + \mathbf{c}_jQe_j && (4.22\text{b}) \end{aligned}$$

with the terminal condition

$$\begin{aligned} \mathbf{S}_j(T) &= \left(\mathbf{c}_jF\mathbf{c}_j^{\mathrm{T}}\right)\big|_{t=T} && (4.23\text{a}) \\ \mathbf{g}_j(T) &= (\mathbf{c}_jFe_j)\big|_{t=T}\ . && (4.23\text{b}) \end{aligned}$$

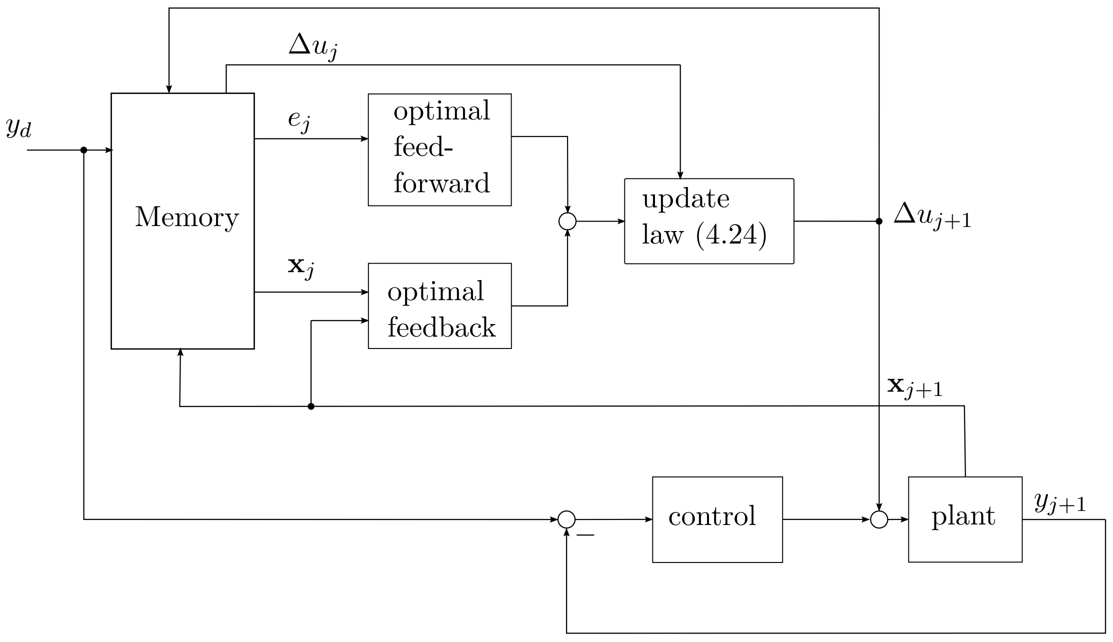


Figure 4.1: Structural diagram LQT-based ILC.

Inserting (4.20) into (4.19a) leads to

$$\Delta u_{j+1} = \Delta u_j - \underbrace{R^{-1}\mathbf{b}_j^{\mathrm{T}}\mathbf{S}_j\Delta\mathbf{z}_{j+1}}_{\text{feedback}} - \underbrace{R^{-1}\mathbf{b}_j^{\mathrm{T}}\mathbf{g}_j}_{\text{feedforward}} \ . \tag{4.24}$$

The only difference between (4.8) and (4.17) is that in (4.8) the system is linearized around the trajectory $(\tilde{\mathbf{x}}, \tilde{u})$ and in (4.17) around $(\tilde{\mathbf{x}}_j, \tilde{u}_j)$. In (4.17), the evolution of the trajectory around which the system gets linearized is given by

$$\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \Delta\mathbf{z}_{j+1} \tag{4.25a}$$
$$\tilde{u}_{j+1} = \tilde{u}_j + \Delta v_{j+1} \ . \tag{4.25b}$$

Implementing this algorithm is equivalent to the implementation of (4.8), except that the linearization around the derived trajectory and solving the Riccati differential equation must be done after every iteration. However, in both approaches it is not *a priori* clear how to choose $(\tilde{\mathbf{x}}, \tilde{u})$ or $(\tilde{\mathbf{x}}_0, \tilde{u}_0)$. An approach to do so could be to solve a trajectory optimization problem or by using a flatness-based strategy, which is related to the next section. Compared to the linear LQT-based ILC from Section 3.1, the AI system has to be stabilized by a controller, which was unnecessary in the linear case since the LQT-based ILC implies the LQT-problem for the zeroth iteration and therefore stabilizes the system. In the nonlinear case, the system is linearized around a given trajectory, and therefore, the system must operate close to this linearization and thus be stabilized. The structural diagram of the algorithm can be seen in Figure 4.1.

4.2 Flatness-based ILC

This section extends the idea that the centrepiece of ILC strategies lies in the model inversion, as presented in Section 3.2 for AI-systems of the form (4.1). Additionally, the algorithm developed within this section drops the need to stabilize the plant with an additional controller like in Section 4.1. Therefore, this approach combines the ILC update law and the controller design within one framework. First, similar to Section 3.2, the relative degree of an AI system is defined in the following definition.

Definition 4.1 (relative degree SISO AI-system [16])**.** *The SISO AI-system (4.1) is said to have the relative degree r at a point $\bar{\mathbf{x}} \in \mathcal{X}$ if*

(i) $\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}) = 0\ ,\qquad \forall i \in [0, r-2]$ *and all* $\mathbf{x}$ *in a neighborhood of* $\bar{\mathbf{x}}$ *,*

(ii) $\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{r-1}h(\bar{\mathbf{x}}) \neq 0\ .$

All system variables of differentially flat systems, including the states and the input variables, can be expressed as functions of a particular output variable and its time derivatives, especially in the SISO case up to order n. Such an output variable is called a *flat output* [16]. As seen in Definition 4.1, a flat output has to fulfill higher-order partial differential equations. As mentioned in [16], the higher-order partial differential equations can be reduced to first-order partial differential equations as follows. First, a recursive form of the Lie brackets [16]

$$\mathrm{ad}_{\mathbf{f}}^{k}\mathbf{g}(\mathbf{x}) = \left[\mathbf{f}, \mathrm{ad}_{\mathbf{f}}^{k-1}\mathbf{g}\right](\mathbf{x})\ , \tag{4.26}$$

with the Lie bracket [16]

$$[\mathbf{f}, \mathbf{g}](\mathbf{x}) = \mathrm{L}_{\mathbf{f}}\mathbf{g}(\mathbf{x}) = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{f} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{g}\ , \tag{4.27}$$

has to be introduced. With the help of these operators and the relationship

$$\mathrm{L}_{[\mathbf{f},\mathbf{g}]}\lambda(\mathbf{x}) = \mathrm{L}_{\mathbf{f}}\mathrm{L}_{\mathbf{g}}\lambda(\mathbf{x}) - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x})\ , \tag{4.28}$$

the higher-order partial differential equations

$$\begin{aligned} \mathrm{L}_{\mathbf{g}}\lambda(\mathbf{x}) &= 0\ ,\\ \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x}) &= 0\ ,\\ &\vdots\\ \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-2}\lambda(\mathbf{x}) &= 0\ ,\\ \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}\lambda(\bar{\mathbf{x}}) &= \alpha \neq 0 \end{aligned} \tag{4.29}$$

can be transformed to first-order partial differential equations of *Frobenius type* [16]

$$\begin{aligned} \mathrm{L}_{\mathbf{g}}\lambda(\mathbf{x}) &= 0\ ,\\ \mathrm{L}_{\mathrm{ad}_{\mathbf{f}}\mathbf{g}(\mathbf{x})}\lambda(\mathbf{x}) &= 0\ ,\\ &\vdots\\ \mathrm{L}_{\mathrm{ad}_{\mathbf{f}}^{n-2}\mathbf{g}(\mathbf{x})}\lambda(\mathbf{x}) &= 0\ ,\\ \mathrm{L}_{\mathrm{ad}_{\mathbf{f}}^{n-1}\mathbf{g}(\mathbf{x})}\lambda(\mathbf{x}) &= \alpha \neq 0\ . \end{aligned} \tag{4.30}$$

Following the presentation in [16] and starting from $\mathrm{L}_{\mathbf{g}}\lambda(\mathbf{x}) = 0$ and $\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x}) = 0$, it directly follows that

$$\mathrm{L}_{\mathrm{ad}_{\mathbf{f}}\mathbf{g}(\mathbf{x})}\lambda(\mathbf{x}) = \mathrm{L}_{\mathbf{f}}\underbrace{\mathrm{L}_{\mathbf{g}}\lambda(\mathbf{x})}_{=0} - \underbrace{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x})}_{=0} = 0\ . \tag{4.31}$$

Therefore, it follows

$$\begin{aligned}\mathrm{L}_{\mathrm{ad}^2_{\mathbf{f}}\mathbf{g}(\mathbf{x})}\lambda(\mathbf{x}) &= \mathrm{L}_{[\mathbf{f},\mathrm{ad}_{\mathbf{f}}\mathbf{g}(\mathbf{x})](\mathbf{x})}\lambda(\mathbf{x}) \\ &= \mathrm{L}_{\mathbf{f}}\underbrace{\mathrm{L}_{\mathrm{ad}_{\mathbf{f}}\mathbf{g}(\mathbf{x})}\lambda(\mathbf{x})}_{=0} - \mathrm{L}_{\mathrm{ad}_{\mathbf{f}}\mathbf{g}(\mathbf{x})}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x}) \\ &= -\left(\mathrm{L}_{\mathbf{f}}\underbrace{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x})}_{=0} - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}\mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x})\right) = \underbrace{\mathrm{L}^2_{\mathbf{f}}\lambda(\mathbf{x})}_{=0} = 0\ .\end{aligned} \tag{4.32}$$

This scheme can be repeated to obtain the remaining relationships of (4.30). The existence of a solution $\lambda(\mathbf{x})$ of (4.30) can be proven by the following Theorem.

Lemma 4.1 ([16] Existence of an output with relative degree $r = n$)**.** *A solution $\lambda(\mathbf{x})$ of the first-order partial differential equations (4.30) in a neighborhood of $\bar{\mathbf{x}}$ exist, if*

(i) the matrix $\left[\mathbf{g}, \mathrm{ad}_{\mathbf{f}}\mathbf{g}, \dots, \mathrm{ad}^{n-1}_{\mathbf{f}}\mathbf{g}\right](\bar{\mathbf{x}})$ has rank n and

(ii) the distribution $D = \{\mathbf{g}, \mathrm{ad}_{\mathbf{f}}\mathbf{g}, \dots, \mathrm{ad}^{n-2}_{\mathbf{f}}\mathbf{g}\}$ is involutive in a neighborhood of $\bar{\mathbf{x}}$.

In this case, the system is called exact input-state linearizable in a neighbourhood of $\bar{\mathbf{x}}$.

Proof. The proof of Lemma 4.1 can be found in [16]. □

If (4.1) satisfies Lemma 4.1, the first-order partial differential equations (4.30) can be solved for $z_1 = \lambda(\mathbf{x})$ and therefore

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \lambda(\mathbf{x}) \\ \mathrm{L}_{\mathbf{f}}\lambda(\mathbf{x}) \\ \vdots \\ \mathrm{L}^{n-1}_{\mathbf{f}}\lambda(\mathbf{x}) \end{bmatrix} = \mathbf{\Phi}(\mathbf{x}) \tag{4.33}$$

is a diffeomorphism. With the help of this transformation, the AI-system from (4.1) can be transformed to

$$\begin{aligned} \dot{z}_1 &= z_2 \\ &\vdots \\ \dot{z}_n &= \mathrm{L}^{n-1}_{\mathbf{f}}\lambda\left(\mathbf{\Phi}^{-1}(\mathbf{z})\right) + \alpha u \\ \mathbf{z}(0) &= \mathbf{z}_0 = \mathbf{\Phi}(\mathbf{x}_0) \in \mathbb{R}^n \\ y &= \lambda\left(\mathbf{\Phi}^{-1}(\mathbf{z})\right)\ . \end{aligned} \tag{4.34}$$

Hence, the states $\mathbf{x}$, the input u, and an arbitrary output y can be parametrized by the flat output z and its derivatives as

$$\begin{aligned} \mathbf{x} &= \mathbf{\Phi}^{-1}(\mathbf{z}) \\ u &= \frac{1}{\alpha}\Big(-\mathrm{L}_{\mathbf{f}}^{n-1}\lambda\Big(\mathbf{\Phi}^{-1}(\mathbf{z})\Big) + z^{(n)}\Big) \\ y &= \lambda\Big(\mathbf{\Phi}^{-1}(\mathbf{z})\Big) \ . \end{aligned} \tag{4.35}$$

This inverse system builds the base to set up a flatness-based ILC.

4.2.1 Optimal tracking problems and the relation to flat systems

First, the dynamical optimization problem

$$\min_{u(\cdot)} \quad J(u) \tag{4.36a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \tag{4.36b}$$

$$y = h(\mathbf{x}) \ , \tag{4.36c}$$

with

$$J(u) = \underbrace{\left(\frac{1}{2}\sum_{i=0}^{n-1} e^{(i)} \sum_{i=0}^{n-1} Q_{i,k} e^{(k)}\right)\Bigg|_{t=T}}_{=\varphi(\mathbf{x}(T))} + \frac{1}{2}\int_0^T \sum_{i=0}^{n-1} e^{(i)} Q_i e^{(i)} + e^{(n)} e^{(n)} \, \mathrm{d}t \tag{4.37}$$

where $Q_i > 0$ and

$$e^{(i)} = y_d^{(i)} - \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \, , \qquad \forall i \in [0, n-1] \tag{4.38a}$$

$$e^{(n)} = y_d^{(n)} - \mathrm{L}_{\mathbf{f}}^{n} h(\mathbf{x}) - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}) u \ . \tag{4.38b}$$

is introduced, where the system is flat concerning the output y.

Theorem 4.2 (Flatness-based control law for an AI-system)**.** *Assume the nonlinear SISO-system (4.1) has a relative degree $r = n$. The unique solution of the optimization problem (4.36) is given by the control law*

$$u = \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})}\left(-\mathrm{L}_{\mathbf{f}}^{n} h(\mathbf{x}) + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x})\Big)\right) \ , \tag{4.39}$$

and the solution of the co-state differential equations

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x})\right)^{\mathrm{T}} \sum_{k=0}^{n-1} Q_{i,k}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x})\Big) \ , \tag{4.40}$$

iff

$$Q_i + 2Q_{i,i-1} - Q_{i,n-1}Q_{n-1,i} = 0 \ , \quad \forall i \in [0, n-1] \tag{4.41a}$$

$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} = 0 \ , \quad \forall i, k \in [0, n-1] \text{ and } i \neq k \tag{4.41b}$$

$$Q_{i,k} = Q_{k,i} \ , \tag{4.41c}$$

with $Q_i > 0$.

Proof. For the optimization problem (4.36), the Hamiltonian function follows as

$$
\begin{aligned}
H = \frac{1}{2}\Bigg[& \sum_{i=0}^{n-1} y_d^{(i)} Q_i y_d^{(i)} - 2\left(\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x})\right)^{\mathrm{T}} Q_i y_d^{(i)} + \left(\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x})\right)^{\mathrm{T}} Q_i \mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}) \\
& + y_d^{(n)} y_d^{(n)} - 2(\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}))^{\mathrm{T}} y_d^{(n)} - 2u\left(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})\right)^{\mathrm{T}} y_d^{(n)} + (\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}))^{\mathrm{T}} \mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}) \\
& - 2(\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}))^{\mathrm{T}} \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}) u + u\left(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})\right)^{\mathrm{T}} \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}) u \Bigg] \\
& + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u)
\end{aligned}
\tag{4.42}
$$

For optimality [14], the following relations must hold in the interval $0 \leq t \leq T$

$$
\dot{\mathbf{x}}^* = \left(\frac{\partial H}{\partial \boldsymbol{\lambda}}(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*)\right)^{\mathrm{T}}, \qquad \mathbf{x}^*(0) = \mathbf{x}_0 \tag{4.43a}
$$

$$
\dot{\boldsymbol{\lambda}}^* = -\left(\frac{\partial H}{\partial \mathbf{x}}(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*)\right)^{\mathrm{T}}, \qquad \boldsymbol{\lambda}^*(T) = \left(\frac{\partial \varphi}{\partial \mathbf{x}_1}\right)^{\mathrm{T}}(\mathbf{x}^*(T)) \tag{4.43b}
$$

$$
0 = \frac{\partial H}{\partial u}(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*) \ . \tag{4.43c}
$$

In the following investigations, the star * for the optimal solution are omitted in the following derivations. From (4.43) it follows

$$
\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \tag{4.44a}
$$

$$
\dot{\boldsymbol{\lambda}} = \sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x})\right)^{\mathrm{T}} Q_i\left(y_d^{(i)} - \mathrm{L}_{\mathbf{f}}^i h(\mathbf{x})\right) \tag{4.44b}
$$

$$
- \left[\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x})\right)^{\mathrm{T}} \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\right]\boldsymbol{\lambda} \tag{4.44c}
$$

$$
- u\left[\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})\right)^{\mathrm{T}} \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\right]\boldsymbol{\lambda} \tag{4.44d}
$$

$$
u = -\frac{1}{(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}))^2}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\lambda} + \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})}(-\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}) + y_d^{(n)}) \ . \tag{4.44e}
$$

In order to find an analytic solution for the co-state differential equation the Ansatz function is introduced as

$$
\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x})\right)^{\mathrm{T}} \sum_{k=0}^{n-1} Q_{i,k}\left(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^k h(\mathbf{x})\right) \ . \tag{4.45}
$$

Inserting the Ansatz function into (4.44d), it follows that

$$\begin{aligned}\dot{\boldsymbol{\lambda}} = &\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}Q_i\left(y_d^{(i)}-\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\\ &+\left[\left(\frac{\partial}{\partial\mathbf{x}}\mathbf{f}(\mathbf{x})\right)^{\mathrm{T}}-\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\right]\\ &\qquad\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)\\ &+u\left[\left(\frac{\partial}{\partial\mathbf{x}}\mathbf{g}(\mathbf{x})\right)^{\mathrm{T}}-\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\right]\\ &\qquad\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)\,.\end{aligned}\tag{4.46}$$

Before continuing the proof, the following identities are introduced.

Lemma 4.3. *The following statements are identical*

$$\left(\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\mathbf{f}(\mathbf{x})\equiv\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i+1}h(\mathbf{x})\right)^{\mathrm{T}}-\left(\frac{\partial}{\partial\mathbf{x}}\mathbf{f}(\mathbf{x})\right)^{\mathrm{T}}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\tag{4.47a}$$

$$\left(\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\mathbf{g}(\mathbf{x})u\equiv u\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}-u\left(\frac{\partial}{\partial\mathbf{x}}\mathbf{g}(\mathbf{x})\right)^{\mathrm{T}}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\tag{4.47b}$$

$$\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})\equiv\mathbf{g}^{\mathrm{T}}(\mathbf{x})\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\,.\tag{4.47c}$$

Proof. In order to proof (4.47a), one has to start with

$$\begin{aligned}\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i+1}h(\mathbf{x})&=\frac{\partial}{\partial\mathbf{x}}\left(\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\mathbf{f}(\mathbf{x})\right)\\ &=\mathbf{f}^{\mathrm{T}}(\mathbf{x})\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})+\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\frac{\partial}{\partial\mathbf{x}}\mathbf{f}(\mathbf{x})\,.\end{aligned}\tag{4.48}$$

The desired result follows from moving the second part of the right-hand side to the left-hand side and transposing both sides. To prove the second identity (4.47b), one has to start with

$$\begin{aligned}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)u&=\left(\frac{\partial}{\partial\mathbf{x}}\left(\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\mathbf{g}(\mathbf{x})\right)\right)u\\ &=u\mathbf{g}^{\mathrm{T}}(\mathbf{x})\left(\frac{\partial^2}{\partial\mathbf{x}^2}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)+\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\left(\frac{\partial}{\partial\mathbf{x}}\mathbf{g}(\mathbf{x})\right)u\,.\end{aligned}\tag{4.49}$$

The desired result follows from moving the second part of the right-hand side to the left-hand side and transposing both sides. In order to prove the third identity (4.47c), the first point of the definition of the relative degree

$$\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})=0\,,\qquad\forall i\in[0,r-2]\,,\tag{4.50}$$

is used. Since $r = n$, it directly follows

$$\mathbf{g}^{\mathrm{T}}(\mathbf{x}) \sum_{i=0}^{n-1} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} = \sum_{i=0}^{n-1} \left(\underbrace{\left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right) \mathbf{g}(\mathbf{x})}_{= \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x})} \right)^{\mathrm{T}} = \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}) \; . \tag{4.51}$$

□

In order to proceed, the Ansatz function (4.45) is differentiated with respect to time, i.e.,

$$\begin{aligned} \dot{\boldsymbol{\lambda}} = &- \sum_{i=0}^{n-1} \left(\frac{\partial^2}{\partial \mathbf{x}^2} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right) (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) u) \sum_{k=0}^{n-1} Q_{i,k} \left(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x}) \right) \\ &- \sum_{i=0}^{n-1} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} \sum_{k=0}^{n-1} Q_{i,k} \left(y_d^{(k+1)} - \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x}) \right) (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) u) \right) \; . \end{aligned} \tag{4.52}$$

Expanding the first summation and using of the identities (4.47a), (4.47b), rearranging the second summation, and inserting the control law u from (4.44e) with the inserted Ansatz function (4.45), it follows

$$\begin{aligned} \dot{\boldsymbol{\lambda}} = &- \sum_{i=0}^{n-1} \left(\left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i+1} h(\mathbf{x}) \right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) \right)^{\mathrm{T}} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} \right) \sum_{k=0}^{n-1} Q_{i,k} \left(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x}) \right) \\ &- u \sum_{i=0}^{n-1} \left(\left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) \right)^{\mathrm{T}} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} \right) \sum_{k=0}^{n-1} Q_{i,k} \left(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x}) \right) \\ &- \sum_{i=0}^{n-1} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} \sum_{k=0}^{n-1} Q_{i,k} \left(y_d^{(k+1)} - \mathrm{L}_{\mathbf{f}}^{k+1} h(\mathbf{x}) \right) \\ &+ \sum_{i=0}^{n-1} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{i} h(\mathbf{x}) \right)^{\mathrm{T}} \underbrace{\sum_{k=0}^{n-1} Q_{i,k} \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{k} h(\mathbf{x})}_{\overset{(4.47c)}{=} Q_{i,n-1} \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})} \Bigg(\\ &\frac{1}{(\mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}))^2} \underbrace{\mathbf{g}^{\mathrm{T}}(\mathbf{x}) \sum_{l=0}^{n-1} \left(\frac{\partial}{\partial \mathbf{x}} \mathrm{L}_{\mathbf{f}}^{l} h(\mathbf{x}) \right)^{\mathrm{T}}}_{\overset{(4.47c)}{=} \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})} \sum_{m=0}^{n-1} Q_{l,m} \left(y_d^{(m)} - \mathrm{L}_{\mathbf{f}}^{m} h(\mathbf{x}) \right) \\ &+ \frac{1}{(\mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}))} \left(-\mathrm{L}_{\mathbf{f}}^{n} h(\mathbf{x}) + y_d^{(n)} \right) \Bigg) \; . \end{aligned} \tag{4.53}$$

Using (4.47c) and rearranging the terms, it finally follows that

$$\begin{aligned}\dot{\boldsymbol{\lambda}} = &\Bigg(\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\Bigg)\\
&\quad\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big)\\
&+u\Bigg(\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x})\Bigg)\\
&\quad\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big)\\
&+\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,n-1}Q_{n-1,k}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big)\\
&-\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=1}^{n-1}Q_{i,k-1}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big)\\
&-\sum_{i=1}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i-1,k}\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big)\end{aligned} \tag{4.54}$$

By subtracting equation (4.46) from (4.54), the coefficients $Q_{i,k}$ can be found. Therefore, the second summation of the second last line of (4.54) is extended to $k = 0$ where $Q_{i,-1} = 0$ and the first summation of the last line to $i = 0$ where $Q_{-1,k} = 0$. By factorizing the corresponding expressions $\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big)$, it follows that

$$\begin{aligned}&-\sum_{\substack{i=0\\ i\neq k}}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}(Q_{i,k-1}+Q_{i-1,k}-Q_{i,n-1}Q_{n-1,k})\Big(y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\Big) \overset{!}{=} 0\\
&-\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}(Q_i+Q_{i,i-1}+Q_{i-1,i}-Q_{i,n-1}Q_{n-1,i})\Big(y_d^{(i)} - \mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\Big) \overset{!}{=} 0\ ,\end{aligned} \tag{4.55}$$

must hold and, therefore

$$Q_i + Q_{i,i-1} + Q_{i-1,i} - Q_{i,n-1}Q_{n-1,i} = 0\ , \quad \forall i \in [0, n-1] \tag{4.56a}$$

$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} = 0\ , \quad \forall i,k \in [0, n-1] \text{ and } i \neq k\ . \tag{4.56b}$$

The next step is to check if the terminal condition is satisfied by the Ansatz function (3.40). For the sake of readability, the time dependency (T) is left away in the next step. To see if the analytic solution matches the final condition, the final cost term is differentiated with respect to the state, i.e.

$$\begin{aligned}
\left(\frac{\partial\varphi}{\partial\mathbf{x}}\right)^{\mathrm{T}} &= \left(\frac{\partial}{\partial\mathbf{x}}\left(\frac{1}{2}\sum_{i=0}^{n-1}\left(y_d^{(i)}-\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)\right)\right)^{\mathrm{T}} &\quad (4.57a)\\
&= \frac{1}{2}\Bigg[-\sum_{i=0}^{n-1}y_d^{(i)}\sum_{k=0}^{n-1}Q_{i,k}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)^{\mathrm{T}} &\quad (4.57b)\\
&\quad -\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)+\sum_{i=0}^{n-1}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\sum_{k=0}^{n-1}Q_{i,k}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)^{\mathrm{T}}\Bigg]\\
&= \frac{1}{2}\Bigg[-\sum_{i=0}^{n-1}\left(y_d^{(i)}-\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)\sum_{k=0}^{n-1}Q_{i,k}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)^{\mathrm{T}} &\quad (4.57c)\\
&\quad -\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)\Bigg]\ . &\quad (4.57d)
\end{aligned}$$

Since $Q_{i,k}=Q_{k,i}$ (see, Theorem 3.1), it finally follows that

$$\left.\left(\frac{\partial\varphi}{\partial\mathbf{x}}\right)^{\mathrm{T}}\right|_{t=T}=\boldsymbol{\lambda}(T)=\left.\left(\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)\right)\right|_{t=T}\ . \quad (4.58)$$

In the last step, (4.45) is inserted into (4.44e) and using (4.47c) one obtains

$$u=\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x})}\left(-\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x})+y_d^{(n)}+\sum_{k=0}^{n-1}Q_{n-1,k}\left(y_d^{(k)}-\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x})\right)\right)\ . \quad (4.59)$$

The calculation of the coefficients $Q_{n-1,k}$ is fully equivalent to the linear case and can be seen in example 1 for $n=1$ and in example 2 for $n=2$. □

From Theorem 4.2, it follows that the flatness-based trajectory following control law is optimal in the sense of minimizing the weighted tracking error given in the cost functional (4.37).

4.2.2 Optimal tracking-based ILC for flat systems

This idea is utilized here in order to build an ILC update law. Therefore, the dynamical optimization problem is formulated in an iterative setting as

$$\begin{aligned}
\min_{u_{j+1}(\cdot)}\quad & J(u_{j+1}) &\quad (4.60a)\\
\text{s.t.}\quad & \dot{\mathbf{x}}_{j+1}=\mathbf{f}(\mathbf{x}_{j+1})+\mathbf{g}(\mathbf{x}_{j+1})u_{j+1} &\quad (4.60b)\\
& y_{j+1}=h(\mathbf{x}_{j+1})\ , &\quad (4.60c)
\end{aligned}$$

with

$$J(u_{j+1}) = \underbrace{\left(\frac{1}{2}\sum_{i=0}^{n-1}(e_{j+1}^{(i)})^{\mathrm{T}}\sum_{i=0}^{n-1}Q_{i,k}e_{j+1}^{(k)}\right)\Bigg|_{t=T}}_{=\varphi(\mathbf{x}_{j+1}(T))} + \frac{1}{2}\int_{t_0}^{t_1}\sum_{i=0}^{n-1}(e_{j+1}^{(i)})^{\mathrm{T}}Q_i(e_{j+1}^{(i)}) + (e_{j+1}^{(n)} - e_j^{(n)})^{\mathrm{T}}(e_{j+1}^{(n)} - e_{j+1}^{(n)})\,\mathrm{d}t \tag{4.61}$$

and

$$e_{j+1}^{(i)} = y_{d,j+1}^{(i)} - \mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1}) \quad , \quad \forall i \in [0, n-1] \tag{4.62a}$$
$$e_{j+1}^{(n)} = y_{d,j+1}^{(n)} - \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j+1}) - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})u_{j+1} \tag{4.62b}$$
$$e_{j}^{(n)} = y_{d,j}^{(n)} - \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_{j}\ . \tag{4.62c}$$

Here, the iteration dependency is formulated fully equivalent to the linear case of Section 3.2.2. Therefore, the desired trajectory $y_{d,j}^{(i)}$ is chosen to be iteration-dependent. Now, it becomes clear why the iteration dependency is chosen in this way. In the nonlinear case, it is impossible to create an iteration-dependent state like in the LQT-based ILC from Section 4.1 and Section 3.2.2. This is because the vector fields cannot be directly compared with each other since they live in different tangent spaces. However, solving this problem leads to the following Theorem.

Theorem 4.4 (Flatness-based ILC law for an AI-system)**.** *Assume the nonlinear SISO-system (4.1) has a relative degree $r = n$. The unique solution of the optimization problem (4.60) is given by the ILC law*

$$u_{j+1} = \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\left(-\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j+1}) + y_d^{(n)} + \sum_{k=0}^{n-1}Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\right)\ , \tag{4.63}$$

with

$$y_{d,j+1}^{(k)} = y_{d,j}^{(k)} + y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_j)\,, \qquad \forall k \in [0, n-1] \tag{4.64}$$

and the solution of the co-state differential equations

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial\mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\ , \tag{4.65}$$

iff

$$Q_i + 2Q_{i,i-1} - Q_{i,n-1}Q_{n-1,i} = 0\ , \quad \forall i \in [0, n-1] \tag{4.66a}$$
$$Q_{i,k-1} + Q_{i-1,k} - Q_{i,n-1}Q_{n-1,k} = 0\ , \quad \forall i,k \in [0, n-1] \text{ and } i \neq k \tag{4.66b}$$
$$Q_{i,k} = Q_{k,i}\ , \tag{4.66c}$$

with $Q_i > 0$.

Proof. For the optimization problem (4.60) the Hamiltonian function follows as

$$\begin{aligned}
H = \frac{1}{2}\Bigg[& \sum_{i=0}^{n-1} y_{d,j+1}^{(i)} Q_i y_{d,j+1}^{(i)} - 2\Big(\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} Q_i y_{d,j+1}^{(i)} + \Big(\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} Q_i \mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1}) \\
& + \Delta y_d^{(n)}\Big(\Delta y_d^{(n)} - 2\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_{j+1}) - 2\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1}) u_{j+1} + 2\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_j) + 2\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j) u_j\Big) \\
& - (\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_{j+1}))^{\mathrm{T}}\Big(-\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1}) - 2\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1}) u_{j+1} + 2\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_j) + 2\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j) u_j\Big) \\
& - u_{j+1}^{\mathrm{T}}\Big(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}}\Big(-\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1}) u_{j+1} + 2\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_j) + 2\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j) u_j\Big) \\
& + (\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_j))^{\mathrm{T}}\Big(\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_j) + 2\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j) u_j\Big) \\
& + u_j^{\mathrm{T}}\Big(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j)\Big)^{\mathrm{T}} \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j) u_j \Bigg] \\
& + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{f}(\mathbf{x}_{j+1}) + \mathbf{g}(\mathbf{x}_{j+1}) u_{j+1}) \ ,
\end{aligned} \tag{4.67}$$

with $\Delta y_d^{(n)} = y_{d,j+1}^{(n)} - y_{d,j}^{(n)}$. By applying the optimality conditions it follows that

$$\dot{\mathbf{x}}_{j+1} = \mathbf{f}(\mathbf{x}_{j+1}) + \mathbf{g}(\mathbf{x}_{j+1}) u_{j+1} \tag{4.68a}$$

$$\dot{\boldsymbol{\lambda}} = \sum_{i=0}^{n-1}\Big(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} Q_i \Big(y_{d,j+1}^{(i)} - \mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1})\Big) \tag{4.68b}$$

$$- \left[\Big(\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} - \Big(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1})}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\right]\boldsymbol{\lambda} \tag{4.68c}$$

$$- u_{j+1}\left[\Big(\frac{\partial}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} - \Big(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x})}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\right]\boldsymbol{\lambda} \tag{4.68d}$$

$$u = -\frac{1}{(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1}))^2}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\boldsymbol{\lambda} \tag{4.68e}$$

$$+ \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1})}(-\mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_{j+1}) + \mathrm{L}_{\mathbf{f}}^n h(\mathbf{x}_j) + \Delta y_d^{(n)} + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_j) u_j) \ . \tag{4.68f}$$

To proceed with the analysis, an Ansatz function for the co-state differential equation, similar to the one used previously, is introduced as

$$\boldsymbol{\lambda} = -\sum_{i=0}^{n-1}\Big(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^i h(\mathbf{x}_{j+1})\Big)^{\mathrm{T}} \sum_{k=0}^{n-1} Q_{i,k}\Big(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^k h(\mathbf{x}_{j+1})\Big) \ . \tag{4.69}$$

Inserting the Ansatz function into (4.68d) leads to

$$
\begin{aligned}
\dot{\boldsymbol{\lambda}} = &\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}} Q_i\left(y_{d,j+1}^{(i)} - \mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)\\
&+\left[\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}_{j+1})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\right]\\
&\qquad\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x})\right)^{\mathrm{T}}\sum_{k=0}^{n-1} Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&+u_{j+1}\left[\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x}_{j+1})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\right]\\
&\qquad\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1} Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\,.
\end{aligned}
\tag{4.70}
$$

Differentiating the Ansatz function (4.69) with respect to time yields

$$
\begin{aligned}
\dot{\boldsymbol{\lambda}} = &-\sum_{i=0}^{n-1}\left(\frac{\partial^2}{\partial \mathbf{x}^2}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)(\mathbf{f}(\mathbf{x}_{j+1}) + \mathbf{g}(\mathbf{x}_{j+1})u_{j+1})\sum_{k=0}^{n-1} Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&-\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1} Q_{i,k}\left(y_{d,j+1}^{(k+1)} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)(\mathbf{f}(\mathbf{x}_{j+1}) + \mathbf{g}(\mathbf{x}_{j+1})u_{j+1})\right)\,.
\end{aligned}
\tag{4.71}
$$

Expanding the first summation and the use of the identities (4.47a), (4.47b), rearranging the second summation, and inserting the control law u from (4.68f), with the inserted

Ansatz function (4.69), it follows

$$
\begin{aligned}
\dot{\boldsymbol{\lambda}} = &-\sum_{i=0}^{n-1}\Bigg(\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i+1}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}} - \bigg(\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\Bigg) \\
&\quad \sum_{k=0}^{n-1} Q_{i,k}\Big(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\Big) \\
&- u_{j+1}\sum_{i=0}^{n-1}\Bigg(\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}} - \bigg(\frac{\partial}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\Bigg) \\
&\quad \sum_{k=0}^{n-1} Q_{i,k}\Big(y_{d}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\Big) \\
&- \sum_{i=0}^{n-1}\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\sum_{k=0}^{n-1} Q_{i,k}\Big(y_{d,j+1}^{(k+1)} - \Big(\mathrm{L}_{\mathbf{f}}^{k+1}h(\mathbf{x}_{j+1})\Big)\Big) \\
&+ \sum_{i=0}^{n-1}\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\underbrace{\sum_{k=0}^{n-1} Q_{i,k}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})}_{\overset{(4.47c)}{=}Q_{i,n-1}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\ \Bigg(\\
&\frac{1}{(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1}))^2}\underbrace{\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\sum_{l=0}^{n-1}\bigg(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{l}h(\mathbf{x}_{j+1})\bigg)^{\mathrm{T}}\sum_{m=0}^{n-1}}_{\overset{(4.47c)}{=}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})} Q_{l,m}\Big(y_{d,j+1}^{(m)} - \mathrm{L}_{\mathbf{f}}^{m}h(\mathbf{x}_{j+1})\Big) \\
&+ \frac{1}{(\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1}))}\Big(-\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j+1}) + \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) + \Delta y_d^{(n)} + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_j\Big)\Bigg)\ .
\end{aligned}
\tag{4.72}
$$

Using (4.47c) and rearranging the terms, it finally follows that

$$\begin{aligned}\dot{\boldsymbol{\lambda}} = &\left(\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}_{j+1})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\right)\\
&\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&+ u_{j+1}\left(\left(\frac{\partial}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x}_{j+1})\right)^{\mathrm{T}} - \left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\mathbf{g}^{\mathrm{T}}(\mathbf{x}_{j+1})\right)\\
&\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&+\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i,n-1}Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&-\sum_{i=0}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=1}^{n-1}Q_{i,k-1}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&-\sum_{i=1}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}\sum_{k=0}^{n-1}Q_{i-1,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\\
&+\sum_{i=1}^{n-1}\left(\frac{\partial}{\partial \mathbf{x}}\mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j+1})\right)^{\mathrm{T}}Q_{i,n-1}\left(\Delta y_{d}^{(n)} - y_{d,j+1}^{(n)} + \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_{j}\right)\end{aligned} \tag{4.73}$$

By taking a closer look at the last part of (4.73), especially at $\Delta y_{d}^{(n)} - y_{d,j+1}^{(n)} + \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_{j}$ and comparing it with (4.70) it follows that

$$\Delta y_{d}^{(n)} - y_{d,j+1}^{(n)} + \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_{j} \overset{!}{=} 0\ , \tag{4.74}$$

in order that the Ansatz function (4.69) is a solution of the optimal co-state differential equation (4.70). Enforcing that $y_{d,j+1}^{(n)} = y_{d}^{(n)}$ leads to

$$\Delta y_{d}^{(n)} = y_{d}^{(n)} - \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_{j} \tag{4.75}$$

and therefore to

$$y_{d,j+1}^{(n)} = y_{d,j}^{(n)} + \Delta y_{d}^{(n)} = y_{d,j}^{(n)} + y_{d}^{(n)} - \mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j}) - \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j})u_{j}\ . \tag{4.76}$$

Using the flat properties from Definition 4.1, it follows that

$$y_{d,j+1}^{(i)} = y_{d,j}^{(i)} + \Delta y_{d}^{(i)} = y_{d,j}^{(i)} + y_{d}^{(i)} - \mathrm{L}_{\mathbf{f}}^{i}h(\mathbf{x}_{j}) \quad , \quad \forall i \in [0, n-1]\ . \tag{4.77}$$

Remark 6. *At this point, it should be noticed that $y_{d,j+1}^{(n)} = y_{d}^{(n)}$ is an assumption since the given trajectory update law exhibits a dead-beat behaviour and converges within one iteration.*

Inserting the Ansatz function (4.69) into (4.68f) leads to

$$u_{j+1} = \frac{1}{\mathrm{L_g L_f^{n-1}} h(\mathbf{x}_{j+1})}\left(-\mathrm{L_f^n} h(\mathbf{x}_{j+1}) + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathrm{L_f^k} h(\mathbf{x}_{j+1})\right)\right) \tag{4.78}$$

The coefficients $Q_{i,k}$ of the Ansatz function (4.69), as well as the matching condition of the final cost-term, can be derived similarly to the proof of Theorem 4.2. □

An intuitive choice for a flatness-based ILC law, similar to the linear case, would be

$$\begin{aligned} u_{j+1} = \frac{1}{\mathrm{L_g L_f^{n-1}} h(\mathbf{x}_{j+1})}\Bigg(& -(\mathrm{L_f^n} h(\mathbf{x}_{j+1}) - \mathrm{L_f^n} h(\mathbf{x}_j)) + y_d^{(n)} + \mathrm{L_g L_f^{n-1}} h(\mathbf{x}_j) u_j \\ & + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_d^{(k)} - \mathrm{L_f^k} h(\mathbf{x}_{j+1})\right)\Bigg) . \end{aligned} \tag{4.79}$$

Therefore, the control law for the zeroth iteration follows as

$$\begin{aligned} u_0 = \frac{1}{(\mathrm{L_g L_f^{n-1}} h(\mathbf{x}_0)}\Bigg(& -\mathrm{L_f^n} h(\mathbf{x}_0) + \underbrace{\mathrm{L_f^n} h(\mathbf{x}_{-1})}_{=y_d^{(n)}} + \mathrm{L_g L_f^{n-1}} h(\mathbf{x}_{-1}) \underbrace{u_{-1}}_{=0} \\ & + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_d^{(k)} - \mathrm{L_f^k} h(\mathbf{x}_0)\right)\Bigg) . \end{aligned} \tag{4.80}$$

Here, $\mathrm{L_f^n} h(\mathbf{x}_{-1}) = y_d^{(n)}$ is chosen in order to get the optimal solution from Section 4.2.1 for the zeroth iteration. However, it should be noted at this point that this is not necessary for the algorithm itself and this is an assumption to the system of the -1-st iteration. For similar reasons, u_{-1} is chosen to be zero. To proceed with the first iteration, it follows

$$\begin{aligned} u_1 = \frac{1}{\mathrm{L_g L_f^{n-1}} h(\mathbf{x}_1)}\Bigg(& -(\mathrm{L_f^n} h(\mathbf{x}_1) + \mathrm{L_f^n} h(\mathbf{x}_0)) + \mathrm{L_g L_f^{n-1}} h(\mathbf{x}_0) u_0 \\ & + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\underbrace{y_d^{(k)}}_{=y_{d,0}^{(k)}} - \mathrm{L_f^k} h(\mathbf{x}_1)\right)\Bigg) , \end{aligned} \tag{4.81}$$

and inserting the zeroth iteration from (4.80), leads to

$$u_1 = \frac{1}{\mathrm{L_g L_f^{n-1}} h(\mathbf{x}_1)}\left(-\mathrm{L_f^n} h(\mathbf{x}_1) + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\underbrace{y_{d,0}^{(k)} + y_d^{(k)} - \mathrm{L_f^k} h(\mathbf{x}_0)}_{=y_{d,1}^{(k)}} - \mathrm{L_f^k} h(\mathbf{x}_1)\right)\right) . \tag{4.82}$$

Continuing this idea to the $j+1$-st iteration, it finally follows

$$u_{j+1} = \frac{1}{\mathrm{L_g L_f^{n-1}} h(\mathbf{x}_{j+1})}\left(-\mathrm{L_f^n} h(\mathbf{x}_{j+1}) + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathrm{L_f^k} h(\mathbf{x}_{j+1})\right)\right) , \tag{4.83}$$

with

$$y_{d,j+1}^{(k)} = y_{d,j}^{(k)} + \underbrace{y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^k h(\mathbf{x}_j)}_{=\bar{e}_j^{(k)}}, \qquad \forall k \in [0, n-1] \; . \tag{4.84}$$

Therefore, the ILC law from (4.78) and (4.79) are equivalent for the given assumptions. From this point, the nature of the ILC can be seen. The so-derived ILC algorithm forms a summation of the trajectory errors over the iteration. In other words, it approximates the mean plant's dynamic behaviour.

4.3 Flatness-based ILC with observer

This section presents a method for the case that not all states of the system can be measured, but only a noisy measurement of the output is available. Therefore, a state observer for the not-measured states has to be used. For this case, the system is assumed in the form

$$\begin{aligned} \dot{\mathbf{x}}_{j+1} &= \mathbf{f}(\mathbf{x}_{j+1}) + \mathbf{g}(\mathbf{x}_{j+1}) u_{j+1} \\ y_{j+1} &= h(\mathbf{x}_{j+1}) \; , \end{aligned} \tag{4.85}$$

where $\mathbf{x}_{j+1} \in \mathbb{R}^n$ is the state and $y_{j+1} \in \mathbb{R}$ is the output. Furthermore, the system is flat concerning the output y_{j+1}. Therefore, the system (4.85) can be transformed into

$$\begin{aligned} \dot{z}_{1,j+1} &= z_{2,j+1} \\ &\vdots \\ \dot{z}_{n,j+1} &= \mathrm{L}_{\mathbf{f}}^n h\Big(\mathbf{\Phi}^{-1}(\mathbf{z}_{j+1})\Big) + \mathrm{L}_{\mathbf{g}} \mathrm{L}_{\mathbf{f}}^{n-1} h\Big(\mathbf{\Phi}^{-1}(\mathbf{z}_{j+1})\Big) u_{j+1} \; , \end{aligned} \tag{4.86}$$

using the diffeomorphism

$$\mathbf{z}_{j+1} = \underbrace{\begin{bmatrix} h(\mathbf{x}_{j+1}) \\ \mathrm{L}_{\mathbf{f}} h(\mathbf{x}_{j+1}) \\ \vdots \\ \mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{x}_{j+1}) \end{bmatrix}}_{=\mathbf{\Phi}(\mathbf{x}_{j+1})} \; . \tag{4.87}$$

Introducing the error in the form

$$\mathbf{z}_{e,j+1} = \mathbf{z}_{j+1} - \underbrace{\begin{bmatrix} y_d \\ y_d^{(1)} \\ \vdots \\ y_d^{(n-1)} \end{bmatrix}}_{=\mathbf{y}_d} \; , \tag{4.88}$$

leads to the observability canonical form (*Brunovsky normal form*)

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} z_{e1,j+1} \\ z_{e2,j+1} \\ \vdots \\ z_{en,j+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} z_{e1,j+1} \\ z_{e2,j+1} \\ \vdots \\ z_{en,j+1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tilde{u}_{j+1} \ , \tag{4.89}$$

with $\tilde{u}_{j+1} = \mathrm{L}_{\mathbf{f}}^{n} h(\mathbf{\Phi}^{-1}(\mathbf{z}_{e,j+1} + \mathbf{y}_d)) + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h(\mathbf{\Phi}^{-1}(\mathbf{z}_{e,j+1} + \mathbf{y}_d))u_{j+1} - y_d^{(n)}$. From this equation, it can be seen that the error dynamics can be written as

$$\dot{\mathbf{z}}_{e,j+1} = \mathbf{A}\mathbf{z}_{e,j+1} + \mathbf{b}\tilde{u}_{j+1} + \mathbf{G}\mathbf{w}_{j+1} \ , \tag{4.90a}$$

$$y_{e,j+1} = \mathbf{c}^{\mathrm{T}}\mathbf{z}_{e,j+1} + v_{j+1} \ , \tag{4.90b}$$

with an additional process noise $\mathbf{w}_{j+1} \in \mathbb{R}^n$ in order to model a stochastic model plant mismatch and the measurement noise $v_{j+1} \in \mathbb{R}$. Assuming the process noise to have zero mean, i.e., $\mathbb{E}(w_{j+1}) = 0$, and the correlation $\mathbb{E}(w_{j+1}(\tau), w_{j+1}(t)) = \mathbf{Q}\delta(t-\tau)$, which is symbolized as $w_{j+1} \sim (0, \mathbf{Q})$. The measurement noise $v_{j+1} \in \mathbb{R}$ is assumed to have zero mean as well, i.e., $\mathbb{E}(v_{j+1}) = 0$, and the correlation $\mathbb{E}(v_{j+1}(\tau), v_{j+1}(t)) = R\delta(t-\tau)$, which is symbolized as $v_{j+1} \sim (0, R)$. The goal is to construct a state observer for the system given in (4.90). Following [18] and [19], one could use a linear observer of the form

$$\dot{\hat{\mathbf{z}}}_{e,j+1} = \mathbf{A}\hat{\mathbf{z}}_{e,j+1} + \mathbf{b}\tilde{u}_{j+1} + \mathbf{k}^f\left(y_{e,j+1} - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{z}}_{e,j+1}\right) \ , \tag{4.91}$$

which is equivalent for a specific choice of $\mathbf{l}(\hat{\mathbf{x}}_{j+1})$ to a nonlinear state observer of the form

$$\dot{\hat{\mathbf{x}}}_{j+1} = \mathbf{f}(\hat{\mathbf{x}}_{j+1}) + \mathbf{g}(\hat{\mathbf{x}}_{j+1})u_{j+1} + \mathbf{l}(\hat{\mathbf{x}}_{j+1})(y_{j+1} - h(\hat{\mathbf{x}}_{j+1})) \ . \tag{4.92}$$

Following the presentation in [19], this can be shown by differentiating the diffeomorphism (4.87) with respect to time, which leads to

$$\dot{\hat{\mathbf{z}}}_{j+1} = \left(\frac{\partial}{\partial \mathbf{x}}\mathbf{\Phi}(\hat{\mathbf{x}}_{j+1})\right)\dot{\hat{\mathbf{x}}}_{j+1} \ . \tag{4.93}$$

By inserting (4.92) it follows that

$$\begin{aligned} \dot{\hat{\mathbf{z}}}_{j+1} = {} & \left(\frac{\partial}{\partial \mathbf{x}}\mathbf{\Phi}(\hat{\mathbf{x}}_{j+1})\right)\mathbf{f}(\hat{\mathbf{x}}_{j+1}) + \left(\frac{\partial}{\partial \mathbf{x}}\mathbf{\Phi}(\hat{\mathbf{x}}_{j+1})\right)\mathbf{g}(\hat{\mathbf{x}}_{j+1})u_{j+1} \\ & + \left(\frac{\partial}{\partial \mathbf{x}}\mathbf{\Phi}(\hat{\mathbf{x}}_{j+1})\right)\mathbf{l}(\hat{\mathbf{x}}_{j+1})(y_{j+1} - h(\hat{\mathbf{x}}_{j+1})) \ , \end{aligned} \tag{4.94}$$

The equivalence of (4.94) and (4.91) can be shown by using Definition 4.1 as follows. Considering each row of (4.94) separately, it follows that

$$\begin{aligned}\dot{\hat{z}}_{1,j+1} &= \underbrace{\left(\frac{\partial}{\partial \mathbf{x}}h(\hat{\mathbf{x}}_{j+1})\right)\mathbf{f}(\hat{\mathbf{x}}_{j+1})}_{=\mathrm{L}_{\mathbf{f}}h(\hat{\mathbf{x}}_{j+1})=\hat{z}_{2,j+1}} + \underbrace{\left(\frac{\partial}{\partial \mathbf{x}}h(\hat{\mathbf{x}}_{j+1})\right)\mathbf{g}(\hat{\mathbf{x}}_{j+1})}_{=\mathrm{L}_{\mathbf{g}}h(\hat{\mathbf{x}}_{j+1})=0} \\ &\quad + \underbrace{\left(\frac{\partial}{\partial \mathbf{x}}h(\hat{\mathbf{x}}_{j+1})\right)\mathbf{l}(\hat{\mathbf{x}}_{j+1})}_{=k_1^f}\left(y_{j+1} - \underbrace{h(\hat{\mathbf{x}}_{j+1})}_{=\hat{z}_{1,j+1}}\right) , \\ \dot{\hat{z}}_{2,j+1} &= \hat{z}_{3,j+1} + k_2^f(y_{j+1} - \hat{z}_{1,j+1}) , \\ &\vdots \\ \dot{\hat{z}}_{n,j+1} &= \mathrm{L}_{\mathbf{f}}^n h(\hat{\mathbf{x}}_{j+1}) + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\hat{\mathbf{x}}_{j+1})u_{j+1} + k_n^f(y_{j+1} - \hat{z}_{1,j+1}) .\end{aligned} \tag{4.95}$$

Using the transformation (4.88), it follows that

$$\begin{aligned}\dot{\hat{z}}_{e1,j+1} &= \hat{z}_{e2,j+1} + k_1^f \underbrace{(y_{j+1} - \hat{z}_{e1,j+1} - y_d)}_{=y_{e,j+1}-\hat{z}_{e1,j+1}} , \\ \dot{\hat{z}}_{e2,j+1} &= \hat{z}_{e3,j+1} + k_2^f(y_{e,j+1} - \hat{z}_{e1,j+1}) , \\ &\vdots \\ \dot{\hat{z}}_{en,j+1} &= \underbrace{\mathrm{L}_{\mathbf{f}}^n h\Big(\mathbf{\Phi}^{-1}(\mathbf{z}_{e,j+1} + \mathbf{y}_d)\Big) + \mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h\Big(\mathbf{\Phi}^{-1}(\mathbf{z}_{e,j+1} + \mathbf{y}_d)\Big)u_{j+1} - y_d^n}_{=\tilde{u}_{j+1}} \\ &\quad + k_n^f(y_{e,j+1} - \hat{z}_{e1,j+1}) .\end{aligned} \tag{4.96}$$

Rewriting (4.96) in vectorized form leads to

$$\dot{\hat{\mathbf{z}}}_{e,j+1} = \mathbf{A}\hat{\mathbf{z}}_{e,j+1} + \mathbf{b}\tilde{u}_{j+1} + \mathbf{k}^f\Big(y_{e,j+1} - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{z}}_{e,j+1}\Big) , \tag{4.97}$$

which is equivalent to (4.92) with the particular choice of the nonlinear observer gain

$$\mathbf{l}(\hat{\mathbf{x}}) = \left(\frac{\partial}{\partial \mathbf{x}}\mathbf{\Phi}(\hat{\mathbf{x}})\right)^{-1}\mathbf{k}^f . \tag{4.98}$$

Remark 7. *Now, it should be clear why the system is transformed into observability canonical form as mentioned above. Using the transformation allows the use of a linear observer and, therefore, the use of linear stochastic optimal observer methods. However, this does not imply stochastic optimality in the nonlinear case since the transformation of Gaussian noise won't be Gaussian in general. Again, the system is transformed into tracking error coordinates (4.88) to receive a system with zero-mean deviation from the nominal system. Additionally, it should be noticed that this observer is an exact linearizing observer.*

The stochastic optimal choice for $\mathbf{k}^f$, is given by a Kalman filter [17], which can be written as

$$\mathbf{k}^f = \mathbf{P}^f \mathbf{c}(R)^{-1} \tag{4.99a}$$

$$\dot{\mathbf{P}}^f = \mathbf{A}\mathbf{P}^f + \mathbf{P}^f \mathbf{A}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} - \mathbf{k}^f R\left(\mathbf{k}^f\right)^{\mathrm{T}} \tag{4.99b}$$

$$\dot{\hat{\mathbf{z}}}^f_{j+1} = \mathbf{A}\hat{\mathbf{z}}^f_{j+1} + \mathbf{b}\tilde{u}_{j+1} + \mathbf{k}^f\left(y_{j+1} - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{z}}^f_{j+1}\right) , \tag{4.99c}$$

with the initial conditions

$$\hat{\mathbf{z}}^f_{j+1}(0) = \mathbf{z}_{j+1}(0) \tag{4.99d}$$

$$\mathbf{P}^f(0) = \mathbf{P}_0 \ . \tag{4.99e}$$

From Theorem 4.4, it follows that the optimal flatness-based ILC, in order to minimize the error up to order n, with full state information, is given by

$$u_{j+1} = \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h(\mathbf{x}_{j+1})}\left(-\mathrm{L}_{\mathbf{f}}^{n}h(\mathbf{x}_{j+1}) + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(y_{d,j+1}^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_{j+1})\right)\right) \tag{4.100}$$

and the trajectory update law as

$$y_{d,j+1}^{(k)} = y_{d,j}^{(k)} + y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k}h(\mathbf{x}_j) , \qquad \forall k \in [0, n-1] \ . \tag{4.101}$$

Since the trajectory update law (4.101) is calculated offline (after every iteration), it only contains quantities from the last iteration j. Therefore, a Kalman smoother is used, which can be written as [17]

$$\mathbf{K}^s = \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}}\left(\mathbf{P}^f\right)^{-1} \tag{4.102a}$$

$$\dot{\mathbf{P}}^s = (\mathbf{A} + \mathbf{K}^s)\mathbf{P}^s + \mathbf{P}^f(\mathbf{A} + \mathbf{K}^s)^{\mathrm{T}} - \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} \tag{4.102b}$$

$$\dot{\hat{\mathbf{z}}}^s_{j+1} = \mathbf{A}\hat{\mathbf{z}}^s_{j+1} + \mathbf{K}^s\left(\hat{\mathbf{z}}^s_{j+1} - \hat{\mathbf{z}}^f_{j+1}\right) , \tag{4.102c}$$

with the final conditions

$$\hat{\mathbf{z}}^s_{j+1}(T) = \hat{\mathbf{z}}^f_{j+1}(T) \tag{4.102d}$$

$$\mathbf{P}^s(T) = \mathbf{P}^f(T) \tag{4.102e}$$

Here, the superscript $(\cdot)^f$ denotes the filter or forward-pass, and $(\cdot)^s$ denotes the smoother. Since the flatness-based ILC algorithm from Section 4.2.2 consists of a feedback and a feedforward part, the estimations of the *Kalman filter* are used for the *feedback* part and the estimations of the *Kalman smoother* for the *trajectory update*. Therefore, (4.100) is implemented as

$$u_{j+1} = \frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1}h\left(\hat{\mathbf{x}}^f_{j+1}\right)}\left(-\mathrm{L}_{\mathbf{f}}^{n}h\left(\hat{\mathbf{x}}^f_{j+1}\right) + y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\hat{y}_{d,j+1}^{(k),s} - \mathrm{L}_{\mathbf{f}}^{k}h\left(\hat{\mathbf{x}}^f_{j+1}\right)\right)\right) , \tag{4.103}$$

with

$$\hat{y}_{d,j+1}^{(k),s} = \hat{y}_{d,j}^{(k),s} + y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h\left(\hat{\mathbf{x}}_j^s\right), \qquad \forall k \in [0, n-1]\ . \tag{4.104}$$

From this point, the observed states of the system can be derived from the observed error state, which is written as

$$\hat{z}_{ie}^{f} = \hat{y}_{j+1}^{(i-1),f} - y_d^{(i-1)} = \mathrm{L}_{\mathbf{f}} h\left(\hat{\mathbf{x}}_{j+1}^f\right) - y_d^{(i-1)}\ , \tag{4.105}$$

an therefore in vectorized form as

$$\hat{\mathbf{z}}_{e,j+1}^{f} = \underbrace{\begin{bmatrix} h\left(\hat{\mathbf{x}}_{j+1}^f\right) \\ \mathrm{L}_{\mathbf{f}}\left(\hat{\mathbf{x}}_{j+1}^f\right) \\ \vdots \\ \mathrm{L}_{\mathbf{f}}^{n}\left(\hat{\mathbf{x}}_{j+1}^f\right) \end{bmatrix}}_{=\boldsymbol{\Phi}\left(\hat{\mathbf{x}}_{j+1}^f\right)} - \underbrace{\begin{bmatrix} y_d \\ y_d^{(1)} \\ \vdots \\ y_d^{(n-1)} \end{bmatrix}}_{=\mathbf{y}_d}\ , \tag{4.106}$$

which leads to

$$\hat{\mathbf{x}}_{j+1}^{f} = \boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j+1}^f + \mathbf{y}_d\right)\ . \tag{4.107}$$

Rewriting (4.103) with the observed states from (4.107) leads to

$$\begin{aligned} u_{j+1} = &\frac{1}{\mathrm{L}_{\mathbf{g}}\mathrm{L}_{\mathbf{f}}^{n-1} h\left(\boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j+1}^f + \mathbf{y}_d\right)\right)}\Bigg(-\mathrm{L}_{\mathbf{f}}^{n} h\left(\boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j+1}^f + \mathbf{y}_d\right)\right) \\ &+ y_d^{(n)} + \sum_{k=0}^{n-1} Q_{n-1,k}\left(\hat{y}_{d,j+1}^{(k),s} - \mathrm{L}_{\mathbf{f}}^{k} h\left(\boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j+1}^f + \mathbf{y}_d\right)\right)\right)\Bigg)\ . \end{aligned} \tag{4.108}$$

For the trajectory update, the Kalman smoother is applied to the error dynamics and therefore

$$\hat{\mathbf{x}}_j^s = \boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j}^s + \mathbf{y}_d\right)\ . \tag{4.109}$$

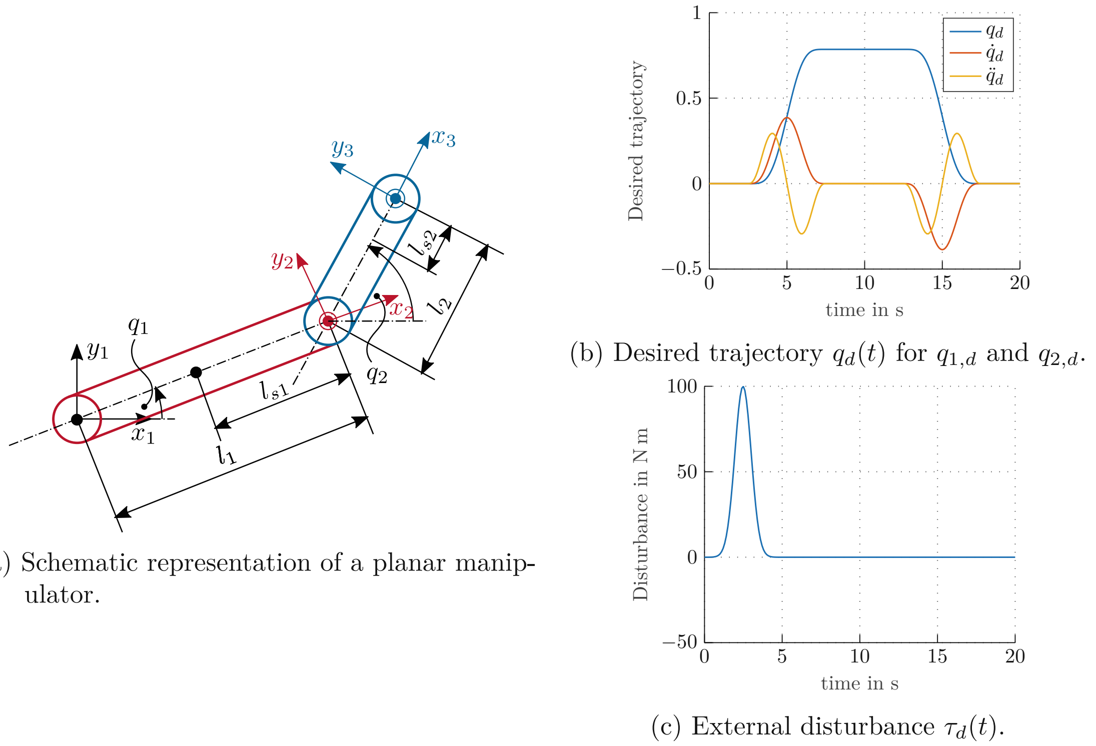
From (4.104) it follows that

$$\hat{y}_{d,j+1}^{(k),s} = \hat{y}_{d,j}^{(k),s} + y_d^{(k)} - \mathrm{L}_{\mathbf{f}}^{k} h\left(\boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j}^s + \mathbf{y}_d\right)\right), \qquad \forall k \in [0, n-1]\ , \tag{4.110}$$

which is written in vectorized form as

$$\hat{\mathbf{y}}_{d,j+1}^{s} = \hat{\mathbf{y}}_{d,j}^{s} + \mathbf{y}_d - \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^{-1}\left(\hat{\mathbf{z}}_{e,j}^s + \mathbf{y}_d\right)\right) = \hat{\mathbf{y}}_{d,j}^{s} - \hat{\mathbf{z}}_{e,j}^{s}\ . \tag{4.111}$$

The ILC-update law (4.108) and the trajectory update (4.111), which is calculated in the offline phase (after every iteration), build up the centrepiece of the flatness-based ILC with an error state observer. However, ILC laws regularly additionally need to deal with repetitive disturbances as well.



(a) Schematic representation of a planar manipulator.

(b) Desired trajectory $q_d(t)$ for $q_{1,d}$ and $q_{2,d}$.

(c) External disturbance $\tau_d(t)$.

Figure 4.2: Schematic representation, desired trajectory, and applied disturbance.

4.4 Example

This section applies the theoretical results presented in this chapter to the example of the planar manipulator shown in Figure 4.2a. The model is derived using the Euler–Lagrange equations. The detailed equations of motion are too extensive to present here in full detail. Nevertheless, after deriving the equations of motion, the system can be written in AI representation of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})(u_1 + \tau_d) + \mathbf{g}_2(\mathbf{x})u_2\,, \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{4.112a}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}\,. \tag{4.112b}$$

Here, the input u_1 of the system is given as the torque τ_1 around the z_1-axis in a clockwise direction and u_2 as the torque τ_2 around the z_2-axis. The external disturbance τ_d acts in addition to τ_1. The system is flat concerning the output of the system y, which represents the generalized coordinates q_1 and q_2. The mechanical parameters can be found in Table 4.1.

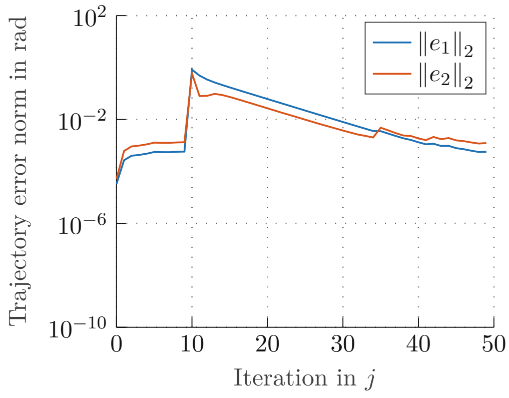
The system and the ILC strategies presented in this chapter were simulated in MATLAB/SIMULINK. The desired trajectory for each iteration can be seen in Figure 4.2b. The disturbance τ_d (see Figure 4.2c) is applied after the 10-th iteration, acting on the system in the same way as u_1. Figure 4.3 shows the L_2 norm of all presented ILC

m_1	$1\,\mathrm{kg}$	m_2	$1\,\mathrm{kg}$
I_1	$1\,\mathrm{kg\,m^2}$	I_2	$1\,\mathrm{kg\,m^2}$
l_1	$1\,\mathrm{m}$	l_2	$1\,\mathrm{m}$
l_{s1}	$0.5\,\mathrm{m}$	l_{s2}	$0.5\,\mathrm{m}$

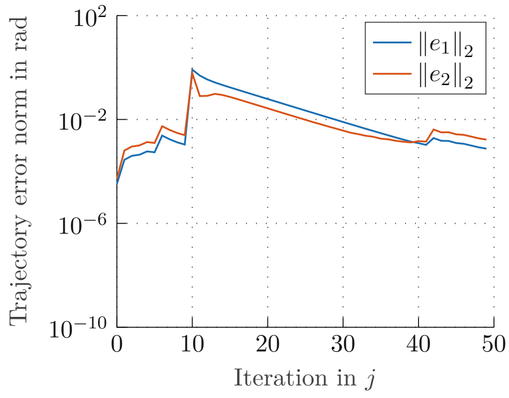
Table 4.1: Parameters of the mathematical model.

strategies. In Figure 4.3a, it can be seen that the LQT-based ILC has a slight increase in the first iterations, which is suspected to stem from the boundary conditions of the optimization problem. It has been shown that the choice of the terminal condition affects the errors occurring during one iteration, which might be because the final conditions are not reachable. However, after the disturbance is applied to the system, the LQT-based ILC has the ability to suppress the error to a level comparable to the undisturbed case. As it can be seen in Figure 4.3b, the LQT-based ILC with trajectory evolution, from (4.17), does not have a direct advantage compared to the formulation without trajectory evolution. In Figure 4.3c, it can be seen that the error converges way faster for the developed flatness-based ILC scheme compared to the other strategies, ultimately reaching a lower value. Figure 4.3d shows the flatness-based ILC with error observer. Here, it can be seen that in the first iterations, before the disturbance is applied, the norm of the error decreases to a value smaller than that of the flatness-based ILC. This is possibly due to numerical errors that occur due to the discrete-time nature of the simulation. These errors could be reduced since the trajectory update has the potential to minimize errors due to the Kalman smoother's ability to estimate states without introducing phase shifts. After the disturbance is applied to the system, the flatness-based ILC with error observer does not converge back to the same value as before applying the disturbance. This is due to the model-plant mismatches in the observer since the disturbance is modelled as a stochastic signal with zero mean. Therefore, in order to deal with this behaviour, one could extend the error observer by a disturbance observer and a first-order Markov process [17] as described in Section 4.3, which leads to the convergence behaviour Figure 4.3e. As it can be seen, the error does not converge back to the value before applying the external disturbance, but the peak value where the disturbance is applied and the final value are lower by approximately the order of two. The fact that the error is not converging back to the value before applying the external disturbance might be due to some remaining unmodelled dynamics in the stochastic process. However, it can be said that convergence behaviour is better compared to the flatness-based ILC without explicitly considering the disturbance in the state estimator. The corresponding control inputs can be seen in Figure 4.4 to show that these comparisons are fair, since it can be seen that they look qualitatively similar for the 0-th, 10-th, and 50-th iteration.

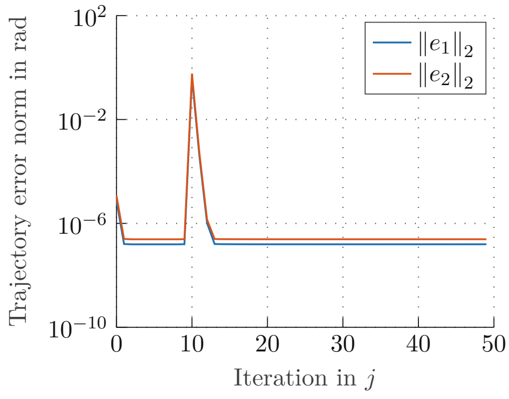
In order to see the ILC strategies under more realistic conditions, Gaussian measurement noise is added with a standard deviation of $1 \cdot 10^{-4}$ rad and zero mean. The convergence behaviour can be seen in Figure 4.5. Compared to the LQT-based ILC before, a zero-phase low-pass filter is required to obtain a stable update law. This zero-phase low-pass filter can be seen as a Q-filter analogous to the discrete-time case and, therefore, is applied to u_j. Again, it can be seen that the flatness-based ILC performs better than the LQT-based



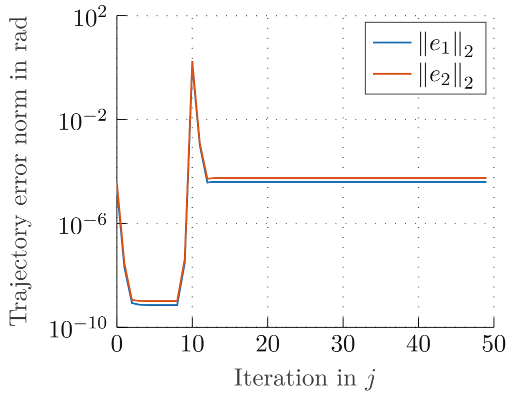
(a) Linearized LQT-based ILC with full state information.



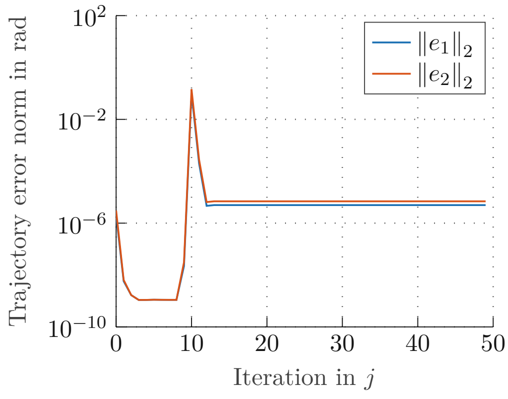
(b) Linearized LQT-based ILC with trajectory evolution and full state information.



(c) Flatness-based ILC with full state information.



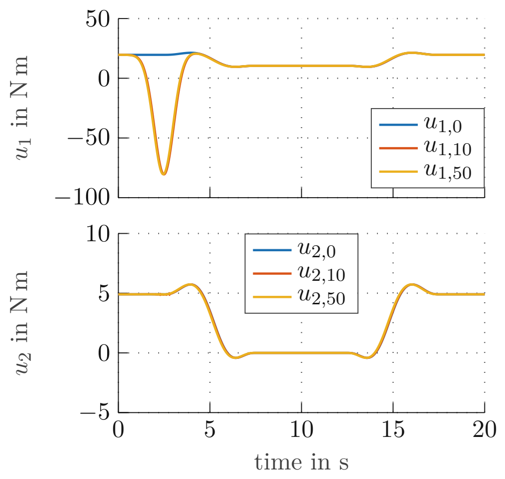
(d) Flatness-based ILC with error dynamics observer.



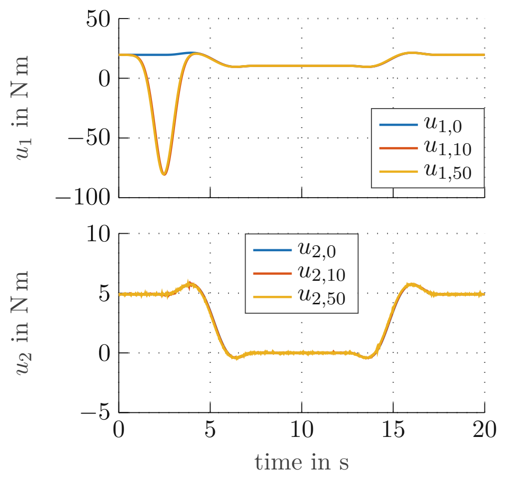
(e) Flatness-based ILC with error dynamics and disturbance observer.

Figure 4.3: Comparison of the convergence behaviour of different ILC algorithms.

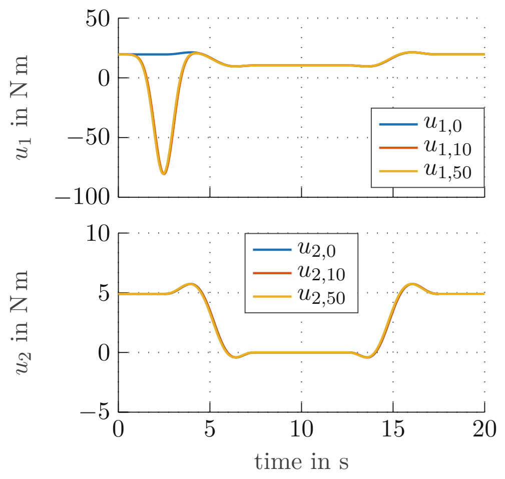
ILC since the error converges within one iteration.



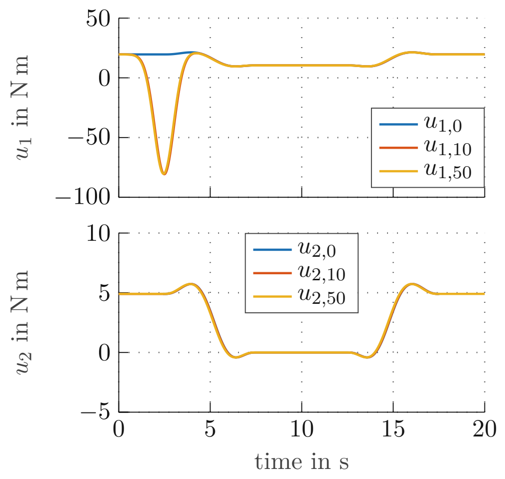
(a) Linearized LQT-based ILC with full state information.



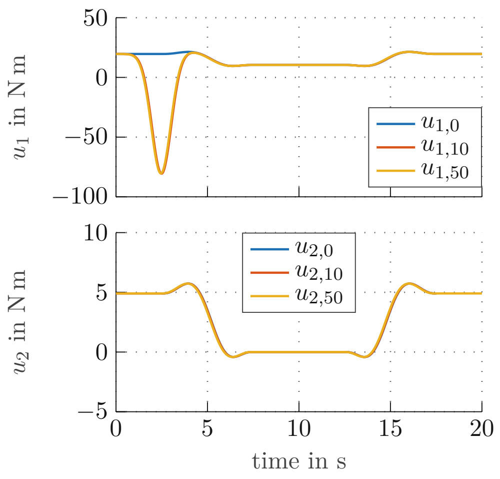
(b) Linearized LQT-based ILC with trajectory evolution and full state information.



(c) Flatness-based ILC with full state information.

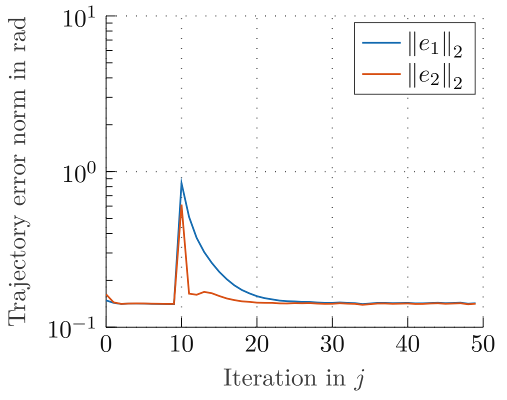


(d) Flatness-based ILC with error dynamics observer.

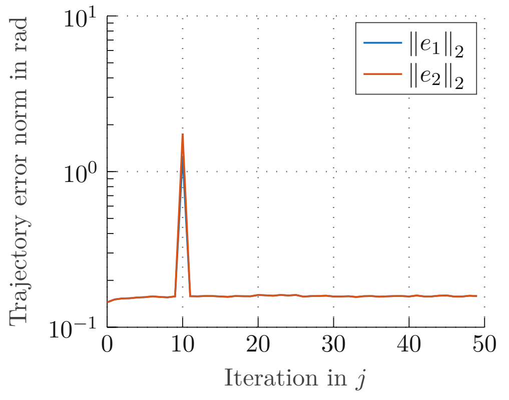


(e) Flatness-based ILC with error dynamics and disturbance observer.

Figure 4.4: Comparison of the control input of different ILC algorithms.



(a) Linearized LQT-based ILC with zero phase filter.



(b) Flatness-based ILC with error dynamics observer.

Figure 4.5: Comparison convergence behaviour with measurement noise.

5 Laboratory helicopter

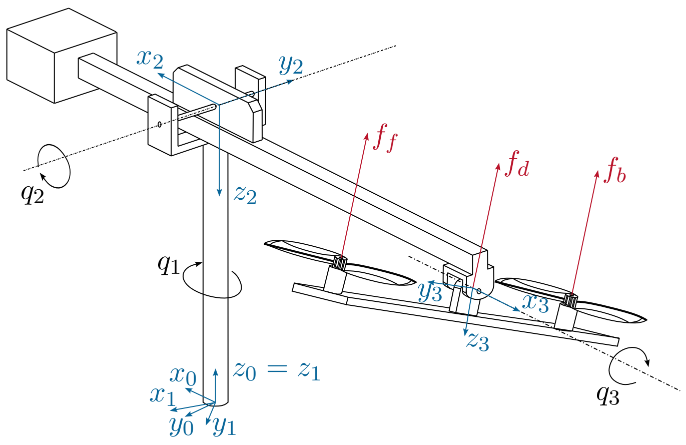


Figure 5.1: Schematic representation of a laboratory helicopter.

This chapter presents the implementation and the results of the flatness-based ILC algorithm developed in Section 4.2 applied to the 3DOF laboratory helicopter from Figure 5.1. A detailed derivation of the mathematical model is given in [20]. The contents of this chapter were essentially, unless otherwise stated, taken from [20–22].

5.1 Mathematical model

As described in [20], the derived model is not *a priori* flat. Therefore, a model reduction is applied, which reproduces the essential dynamics and nonlinearities. This leads to the following AI-system

$$\frac{\mathrm{d}}{\mathrm{d}t}\underbrace{\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix}x_2\\0\\x_4\\f_2(x_3)\\x_6\\f_3(x_3,x_5)\end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix}0\\g_1(x_3,x_5)\\0\\g_2(x_5)\\0\\0\end{bmatrix}}_{\mathbf{g}_1(\mathbf{x})} u_1 + \underbrace{\begin{bmatrix}0\\0\\0\\0\\0\\g_3\end{bmatrix}}_{\mathbf{g}_2(\mathbf{x})} u_2 \tag{5.1a}$$

$$\mathbf{y} = \begin{bmatrix}x_1\\x_3\end{bmatrix} = \mathbf{h}(\mathbf{x}) \tag{5.1b}$$

a_1	$0.3496\,\mathrm{rad/s^2}$	b_1	$-0.6333\,\mathrm{rad/(kg\,m)}$
a_2	$-1.1586\,\mathrm{rad/s^2}$	b_2	$-0.6501\,\mathrm{rad/(kg\,m)}$
a_3	$-0.9035\,\mathrm{rad/s^2}$	b_3	$4.6244\,\mathrm{rad/(kg\,m)}$

Table 5.1: Parameters of the mathematical model [21].

with

$$f_2(x_3) = a_1 \sin x_3 + a_2 \cos x_3 \tag{5.2a}$$
$$f_3(x_3, x_5) = a_3 \cos x_3 \sin x_5 \tag{5.2b}$$
$$g_1(x_3, x_5) = b_1 \cos x_3 \sin x_5 \tag{5.2c}$$
$$g_2(x_5) = b_2 \cos x_5 \tag{5.2d}$$
$$g_3 = b_3 \tag{5.2e}$$

and the transformed inputs $u_1 = f_f + f_b$ and $u_2 = f_f - f_b$. The geometry-dependent parameters a_i and b_i, $i = 1, 2, 3$ are given in Table 5.1.

5.2 Quasi-static state feedback

Following [23] and [20], a quasi-static state feedback for the system (5.1) of the form

$$u_1 = \chi_1(y_{d2}, \dot{y}_{d2}, \ddot{y}_{d2}, y_2, \dot{y}_2, y_3, e_{2I}) \tag{5.3a}$$
$$u_2 = \chi_2\left(y_{d1}^{(i)}, y_{d2}^{(i)}, y_1, \dot{y}_1, y_2, \dot{y}_2, y_3, \dot{y}_3, e_{1I}, e_{2I}\right), \qquad \forall i \in [0, 4] \tag{5.3b}$$

can be found by choosing the error dynamics as

$$y_1^{(4)} = y_{d1}^{(4)} + \sum_{i=0}^{3} Q_{1,(i+1)}\left(y_{d1}^{(i)} - y_1^{(i)}\right) + Q_{1,0} \underbrace{\int (y_{d1} - y_1)\,\mathrm{d}t}_{=e_{1I}} \tag{5.4a}$$

$$y_2^{(2)} = y_{d2}^{(2)} + \sum_{i=0}^{1} Q_{2,(i+1)}\left(y_{d2}^{(i)} - y_2^{i}\right) + Q_{2,0} \underbrace{\int (y_{d2} - y_2)\,\mathrm{d}t}_{=e_{2I}} . \tag{5.4b}$$

Applying the idea of Theorem 4.4, the quasi-static state feedback can be utilized to obtain an ILC law by choosing

$$\hat{y}_{1,j+1}^{(4),f} = \hat{y}_{d1,j+1}^{(4),s} + \sum_{i=0}^{3} Q_{1,(i+1)}\left(\hat{y}_{d1,j+1}^{(i),s} - \hat{y}_{1,j+1}^{(i),f}\right) + Q_{1,0} \int \underbrace{\left(\hat{y}_{d1,j+1}^{s} - \hat{y}_{1,j+1}^{f}\right)}_{=\hat{e}_{1I,j+1}} \mathrm{d}t \tag{5.5a}$$

$$\hat{y}_{2,j+1}^{(2),f} = \hat{y}_{d2}^{(2),s} + \sum_{i=0}^{1} Q_{2,(i+1)}\left(\hat{y}_{d2,j+1}^{(i),s} - \hat{y}_{1,j+1}^{(i),f}\right) + Q_{2,0} \int \underbrace{\left(\hat{y}_{d2,j+1}^{s} - \hat{y}_{2,j+1}^{f}\right)}_{=\hat{e}_{2I,j+1}} \mathrm{d}t \ , \tag{5.5b}$$

and therefore, the quasi-static state feedback-based ILC with an error observer is given by

$$
\begin{aligned}
u_{1,j+1} &= \chi_1\left(\hat{y}^s_{d2,j+1}, \dot{\hat{y}}^s_{d2,j+1}, \ddot{\hat{y}}^s_{d2,j+1}, \hat{y}^f_{2,j+1}, \dot{\hat{y}}^f_{2,j+1}, \hat{y}^f_{3,j+1}, \hat{e}_{2I,j+1}\right) && \text{(5.6a)}\\
u_{2,j+1} &= \chi_2(\hat{y}^{(i),s}_{d1,j+1}, \hat{y}^{(i),s}_{d2,j+1}, \hat{y}^f_{1,j+1}, \dot{\hat{y}}^f_{1,j+1}, \hat{y}^f_{2,j+1}, \dot{\hat{y}}^f_{2,j+1}, \hat{y}^f_{3,j+1}, \dot{\hat{y}}^f_{3,j+1},\\
&\qquad \hat{e}_{1I,j+1}, \hat{e}_{2I,j+1})\ , \quad \forall i \in [0,4] && \text{(5.6b)}
\end{aligned}
$$

with the trajectory update law

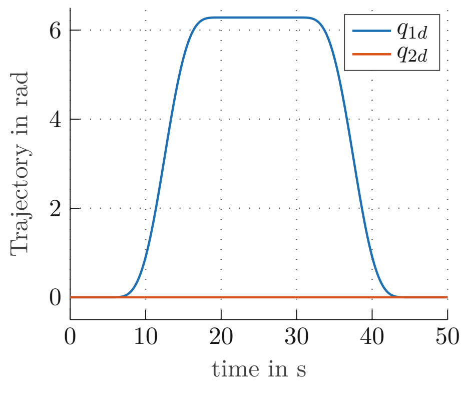
$$
\begin{aligned}
\hat{y}^{(i)}_{d1,j+1} &= \hat{y}^{(i),s}_{d1,j} + y^{(i)}_{d1} - \hat{y}^{(i),s}_{1,j}\ , && \forall i \in [0,3] && \text{(5.7a)}\\
\hat{y}^{(i)}_{d2,j+1} &= \hat{y}^{(i),s}_{d2,j} + y^{(i)}_{d2} - \hat{y}^{(i),s}_{2,j}\ , && \forall i \in [0,1]\ . && \text{(5.7b)}
\end{aligned}
$$

The outputs y_1 and y_2 are estimated with the Kalman filter $(\cdot)^f$, Kalman smoother $(\cdot)^s$, and the shaping filter as a first-order Markov process as derived in Section 4.3. The quasi-static feedback ILC (5.6) has to be implemented, which deals with the information of the current iteration and, therefore, is denoted as the feedback part. The trajectory update law (5.7) is calculated during the offline phase (after every iteration), since it only deals with information of the last iteration. Similar to the flatness-based ILC with state observer from Section 4.3, the Kalman filter estimates the states, and therefore the output, for the feedback part of the control law, and the Kalman smoother is used to estimate the state (output) for the trajectory update law.

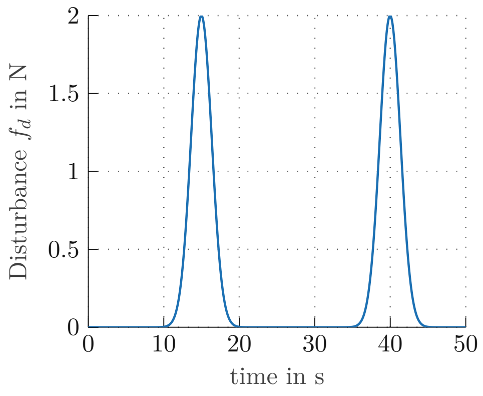
5.3 Simulation results

This section shows the simulation results of the laboratory helicopter in MATLAB/SIMULINK with the ILC concepts designed in Section 5.2. In order to show the algorithm's performance, two test scenarios are investigated. In the first test, the elevation angle q_2 is set to be zero. Therefore, the reduced model and the model derived from the Euler-Lagrange formalism match quite well [23]. Since the model reduction is quite accurate for $q_2 \approx 0$, the second test scenario was chosen to increase those model discrepancies by varying q_2 as well. Additionally, it has been shown in [23] that the system is only flat for $q_2 = 0$ and therefore, this test scenario should show the power of the presented ILC algorithms.

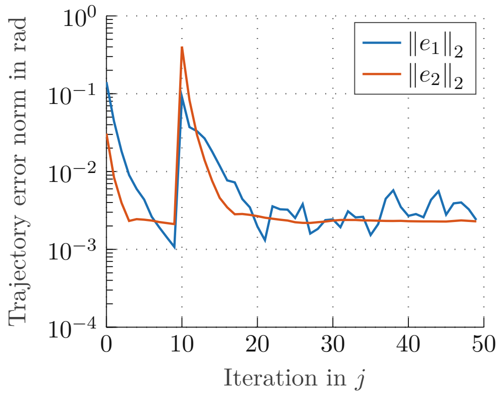
Figure 5.2 shows the results for the first test scenario. For this test case, the desired trajectory for each iteration can be seen in Figure 5.2a and the disturbance f_d, which is applied after the 10-th iteration, can be seen in Figure 5.2b. The external disturbance f_d is directly acting on the elevation angle as presented in Figure 5.1. As it can be seen in Figure 5.2c, the ILC strategy from Section 5.2 can deal with the model plant mismatches from the model reduction as well as with the disturbance. Figure 5.3 shows the results for the second test scenario. Figure 5.3b shows the convergence behaviour for this case, which leads to qualitatively similar results. Therefore, the quasi-static feedback ILC can deal with the violation of the flatness condition that q_2 must be zero.



(a)



(b)

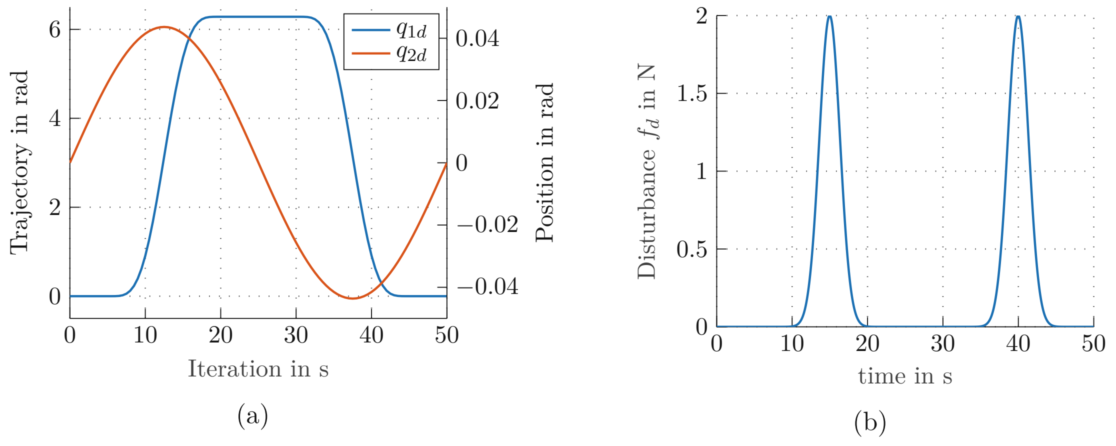


(c)

Figure 5.2: a) Desired trajectory, b) applied disturbance $f_d(t)$ in N, and c) convergence behaviour for the first test case.

5.4 Measurement results

This section presents the measurement results of the ILC presented in Section 5.2 applied to the laboratory helicopter of the company Quanser. A schematic representation of the measurement setup is given in Figure 5.4a and the desired trajectory in Figure 5.4b. Two different test scenarios are analyzed to test the performance of the ILC presented in the real setup. In the first scenario, a disturbance is applied to the helicopter in the form of a strong airflow in a specific region of the helicopter's trajectory, which is presented Figure 5.4a. Since the helicopter flies through this external disturbance repeatedly, it can be approximately considered a repetitive disturbance process. Additionally, compared to the simulation test scenarios, this disturbance acts on the helicopter in such a way, that it additionally applies directly to q_3. For this case, the convergence behaviour is demonstrated in Figure 5.5a, where the grey shaded area marks the area with active disturbance. It can be seen that before the disturbance is applied, the norm of the error decreases. Afterwards, the value converges again approximately to the value without disturbance. The learned behaviour increases the error after turning off the external



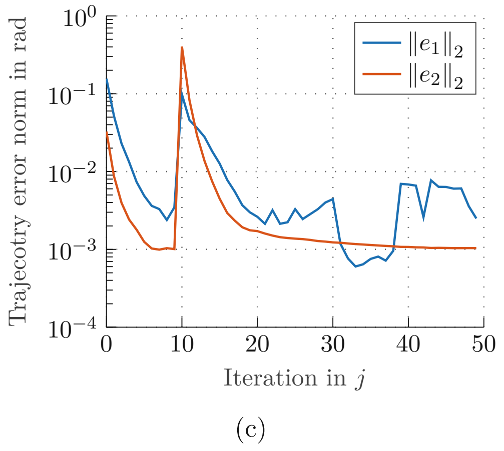
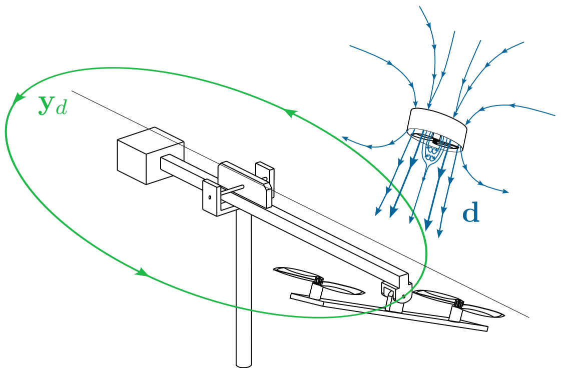
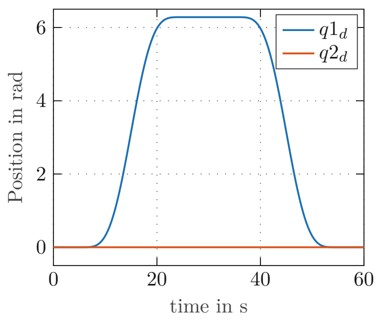


Figure 5.3: a) Desired trajectory, b) applied disturbance $f_d(t)$ in N, and c) convergence behaviour for the second test case.

disturbance, which is then compensated again. In the second scenario, the external fan is positioned below the trajectory plane for the second test scenario. This test reflects a more critical scenario since the rotors of the laboratory helicopter cannot apply downward thrust. Nevertheless, Figure 5.5b shows a similar qualitative behaviour. The norm of the error before applying the disturbance is higher in this case due to other learning coefficients. Figure 5.6 shows the measured tracking error for the 0-th, 19-th, and 30-th iteration for the first test case. Focusing on Figure 5.6a, it can be seen that the initial matching condition is violated for the 19-th iteration, the ILC law is still leading to a stable trajectory update, which mirrors the robustness of the presented algorithm. Since the desired trajectory for the elevation angle is set to be zero for the first test scenario, the accuracy of the algorithm can be seen in Figure 5.6b. Here it can be seen, that the trajectory update reduces the error down to a few encoder steps. Additionally, the location of the external disturbance can be seen approximately at $t = 18\,$s and $t = 43\,$s. There are still residual errors, which might be due to non-repetitive turbulances.

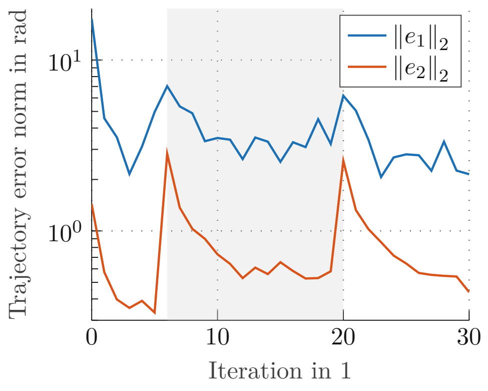


(a) Schematic representation of a laboratory helicopter with external disturbance.

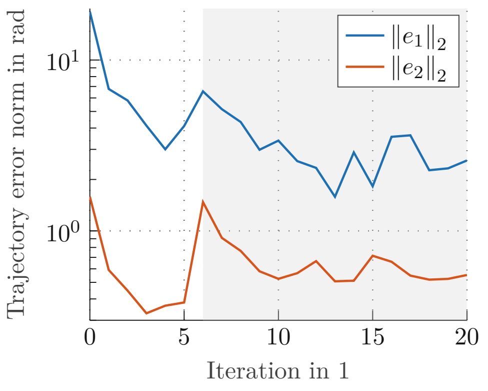


(b) Desired trajectory.

Figure 5.4: Schematic measurement setup and desired trajectory.

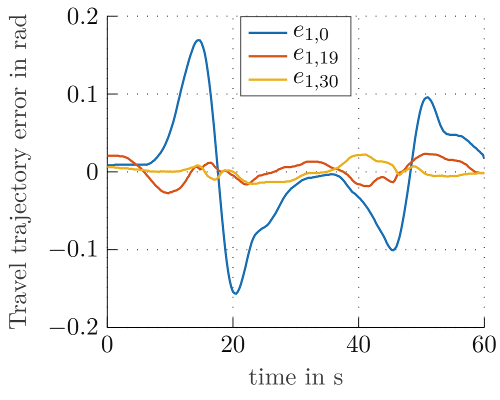


(a) Test case 1.

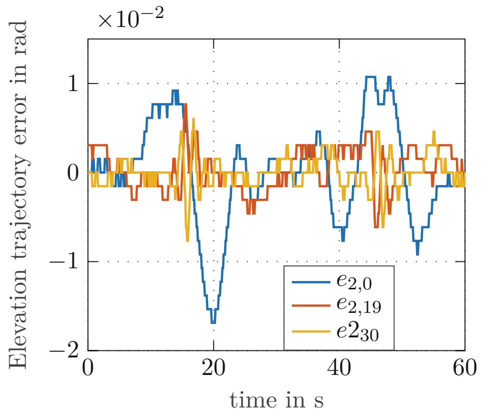


(b) Test case 2.

Figure 5.5: Measured convergence behaviour.



(a) Travel tracking error.



(b) Elevation tracking error.

Figure 5.6: Measured tracking error for the 0-th, 19-th, and 30-th iteration for the first test case.

6 Conclusions

In this thesis, methods of iterative learning control (ILC) for nonlinear systems in affine input (AI) form were investigated. First, the existing theory of ILC for linear systems was presented. Furthermore, an optimization-based ILC framework was developed specifically for linear continuous-time systems, focusing on the properties of differentially flat systems. An analytical solution for the differential equation of the co-state was found. It was also shown that an analytical solution can be constructed for a specific type of Riccati differential equation. These results can be used to develop an efficient method for computing optimal learning laws. Finally, the resulting framework was adapted to nonlinear AI-systems, extending its applicability to more complex control scenarios. For both the linear and nonlinear case, the framework has been extended by integrating both causal and non-causal state estimators. This novel approach enables the systematic design of ILC schemes that capture and utilize the nonlinearities of the plant by tackling them directly.

Although the theoretical derivations were only shown in the single-input single-output (SISO) case, the results can be extended to the multiple-input multiple-output (MIMO) case, which was demonstrated by applying them to the laboratory helicopter of the company Quanser.

There are still many open questions that need to be answered. To point out a few, it remains unclear if and how the framework can be extended to non-flat systems. Furthermore, for the proof of the analytical solution, the underlying system has been assumed to be perfect and, therefore, without external disturbance, process noise, and measurement noise. Therefore, it remains unclear if and how it is possible to systematically take these into account and formulate a rigorous proof. In addition, several key issues need to be clarified for a rigorous stability and robustness analysis of the methods. First, rigorous stability criteria must be established for the ILC algorithms in linear and nonlinear cases with uncertainties. Necessary conditions for convergence in the presence of disturbances need to be determined. Furthermore, it is crucial to investigate how optimal learning rates can be systematically determined in order to achieve a balance between convergence speed and robustness. It is also worthwhile to investigate the impact of Q-filter designs on the performance of ILC methods. However, even if there is no rigorous proof for the analytic solution of the co-state differential equation in the presence of model-plant mismatches, external disturbances, and noise, the result might be additionally interesting in a model predictive control (MPC) setting.

Bibliography

[1] A. Steinhauser, "Optimal iterative learning control for mechatronic systems," Ph.D. dissertation, KU Leuven, 2019.

[2] O. Koçan, "Feedback via Iterative Learning Control for Repetitive Systems," Ph.D. dissertation, Université de Toulouse, 2020.

[3] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[4] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of Robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[5] N. Katoh, K. Nakao, and M. Hanawa, "Learning control of a batch reactor," *Computers & Chemical Engineering*, vol. 13, no. 11, pp. 1273–1276, 1989.

[6] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.

[7] N. Sahoo, J. Xu, and S. Panda, "Low torque ripple control of switched reluctance motors using iterative learning," *IEEE Transactions on Energy Conversion*, vol. 16, no. 4, pp. 318–326, 2001.

[8] R. Kulawinek, K. Galkowski, L. Grzesiak, and A. Kummert, "Iterative learning control method for a single-phase inverter with sinusoidal output voltage," in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, 2011, pp. 1402–1407.

[9] R. Wei, S. Balasubramanian, L. Xu, and J. He, "Adaptive iterative learning control design for rupert iv," in *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2008, pp. 647–652.

[10] Shou-Han Zhou, D. Oetomo, Ying Tan, E. Burdet, and I. Mareels, "Modeling Individual Human Motor Behavior Through Model Reference Iterative Learning Control," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 7, pp. 1892–1901, 2012.

[11] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative Learning Control: Brief Survey and Categorization," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1099–1121, 2007.

[12] D. H. Owens, *Iterative Learning Control*. London, 2016.

[13] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications*. New York, 2007.

[14] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*. 2012.

[15] YangQuan Chen and K. Moore, "An optimal design of PD-type iterative learning control with monotonic convergence," in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, 2002, pp. 55–60.

[16] A. Isidori, *Nonlinear Control Systems*. London, 1995.

[17] F. L. Lewis, L. Xie, and D. Popa, "Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory," 2017.

[18] G. G. Rigatos, "A Derivative-Free Kalman Filtering Approach to State Estimation-Based Control of Nonlinear Systems," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 10, pp. 3987–3997, 2012.

[19] G. G. Rigatos, *Nonlinear Control and Filtering Using Differential Flatness Approaches: Applications to Electromechanical Systems*. 2015.

[20] T. Kiefer, A. Kugi, and W. Kemmetmüller, "Modeling and flatness-based control of a 3dof helicopter laboratory experiment," *IFAC Proceedings Volumes*, vol. 37, no. 13, pp. 207–212, 2004.

[21] T. Kiefer, A. Kugi, K. Graichen, and M. Zeitz, "Feedforward and Feedback Tracking Control of a 3DOF Helicopter Experiment under Input and Output Constraints," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 1586–1593.

[22] T. Kiefer, K. Graichen, and A. Kugi, "Trajectory Tracking of a 3DOF Laboratory Helicopter Under Input and State Constraints," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 944–952, 2010.

[23] A. Kugi and T. Kiefer, "Nichtlineare Trajektorienfolgeregelung für einen Laborhelikopter," *e&i Elektrotechnik und Informationstechnik*, vol. 122, no. 9, pp. 300–307, 2005.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct - Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Dezember 2024

Raphael Buchinger